Interior Exploration Using Seismic Investigations, Geodesy, and Heat Transport (InSight) Mission

Instrument Deployment Arm (IDA)

PDS Archive

Software Interface Specification

Rev. 3.3

December 21, 2020

Prepared by

Susan Slavney, PDS Geosciences Node, Washington University in St. Louis, susan.slavney@wustl.edu

Hallie Abarca, InSight MIPL System Engineer, JPL, hallie.e.gengl@jpl.nasa.gov

Eloise Marteau, InSight IDA Investigation Scientist, JPL, eloise.marteau@jpl.nasa.gov

Grace Lim, Payam Zamani, Galen Hollins, Nythi Udomkesmalee, Cristina De Cesare, Anna Waldron, JPL

InSight

Instrument Deployment Arm

PDS Archive

Software Interface Specification

Rev. 3.3

December 21, 2020

Custodians:

____________________________________ Hallie Abarca InSight MIPL System Engineer	____________ Date
____________________________________ Eloise Marteau InSight IDA Investigation Scientist	____________ Date

Approved:

____________________________________ Ashitey Trebi-Ollenu InSight IDA Principal Investigator	____________ Date
____________________________________ Raymond E. Arvidson PDS Geosciences Node Manager	____________ Date

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

This software interface specification (SIS) describes the format and content of the Instrument Deployment Arm (IDA) Planetary Data System (PDS) data archive. It includes descriptions of the data products and associated metadata, and the archive format, content, and generation pipeline.

1.1 Document Change Log

Table 1 - Document Change Log

DATE	SECTIONS CHANGED	REVISION
2014-04-23	All	V0.0
2015-04-06	All	V0.3
2017-06-07	Section 2, inputs provided by Khaled Ali	V0.4
2017-12-04	6.3: Changed description for Coordinated Frame Origin	V1.1
2019-01-08	TBD Items; Acronyms; 1.3: Applicable Documents, 3.3 Data Validation; 4.2 Label and Header Descriptions; 6.2 Time Standards; 7. IDS Archive Organization (new); 8.2 Applicable PDS Software Tools; plus marginal comments throughout	V1.2
2019-02-27	RDR added to title; 5.0 section title changed to PDS Data Product Specifications; text added to 5.2, 5.3, and 5.4 to clarify which products go to PDS	V1.2
2019-05-31	Extensive updates throughout the document	V2.0
2019-09-12	Revisions after peer review	V2.2
2020-05-11	Section 5.4.1 updated	V3.1
2020-09-10	5.1.4. IDA Position in Spacecraft coordinates (new); 5.1.5 Description of IDA calibration algorithms	V3.2
2020-12-21	Appendix D – End effector forces model	V3.3

1.2 TBD Items

Table 2 - List of TBD items

Section	Item	Who
Section 2.3	Soil Mechanics Calibration Report	Eloise Marteau

1.3 Abbreviations

Table 3 - Abbreviations and their meanings

Abbreviation	Meaning
A/D	Analog-to-Digital

AMMOS	Advanced Multi-Mission Operations System
APID	Application Identifier
APPS	AMMOS-PDS Pipeline Service
APSS	Auxiliary Payload Sensor Subsystem
ASCII	American Standard Code for Information Interchange
CCSDS	Consultative Committee for Space Data Systems
C&DH	Command and Data Handling
DEM	Digital Elevation Model
DH	Denavit and Hartenberg
DSN	Deep Space Network
EDR	Experiment Data Record
FOV	Field of View
FSW	Flight Software
GDS	Ground Data System
HOP	High Input Paraffin
HP^3	Heat Flow and Physical Properties Package
ICC	InSight Context Camera
IDC	InSight Deployment Camera
IDA	InSight Deployment Arm
IDPH	Image Data Product Header
IDS	InSight Deployment System
IPIC	InSight Payload Interface Card
JPL	Jet Propulsion Laboratory
LID	Logical Identifier
LIDVID	Versioned Logical Identifier (logical identifier with version identifier)
LSA	Load Shunt Assembly
LSB	Least Significant Byte
MC	Motor Controller
MGA	Medium Gain Antenna
MIPL	Multimission Instrument Processing Lab
N/A	Not Applicable
NASA	National Aeronautics and Space Administration
NSSDC	National Space Science Data Center
NSYT	Acronym for the “InSight” project
PDS	Planetary Data System
PDS4	Planetary Data System Version 4
PEB	Payload Electronics Box

RA	Robotic Arm
RAE	Robotic Arm Electronics
RDR	Reduced Data Record
RISE	Rotational and Interior Structure Experiment
RSVP	Rover Sequencing and Visualization Program
SCLK	Spacecraft Clock
SEIS	Seismic Experiment for Investigating the Subsurface
SFDU	Standard Format Data Unit
SIS	Software Interface Specification
SOL	Mars Solar day
SPEX	Spectropolarimeter for Planetary Exploration
SPICE	Spacecraft, Planet, Instrument, C-matrix, Events kernels
TBD	To Be Determined/Defined
TBPB	To Be Provided By
TDS	Telemetry Delivery Subsystem
TWINS	Temperature and Wind for InSight
UHF	Ultra High Frequency
UTC	Coordinated Universal Time
VICAR	Video Image Communication and Retrieval
VID	Version Identifier
WTS	SEIS wind and temperature shield
XML	Extensible Markup Language

1.4 Glossary

Many of these definitions are taken from Appendix A of the PDS4 (Planetary Data System Version 4) Concepts Document, pds.nasa.gov/pds4/doc/concepts. The reader is referred to that document for more information.

Archive – A place in which public records or historical documents are preserved; also the material preserved, often used in plural. The term may be capitalized when referring to all of PDS holdings (i.e., the PDS Archive).

Basic Product – The simplest product in PDS4; one or more data objects (and their description objects), which constitute (typically) a single observation, document, etc. The only PDS4 products that are *not* basic products are collection and bundle products.

Bundle – A list of related collections. For example, a bundle could list a collection of raw data obtained by an instrument during its mission lifetime, a collection of the calibration products associated with the instrument, and a collection of all documentation relevant to the first two collections.

Class – The set of attributes (including a name and identifier) which describes an item defined in the PDS Information Model. A class is generic, i.e., a template from which individual items may be constructed.

Collection – A list of closely related basic products of a single type (e.g. observational data, browse files, documents, etc.). A collection is itself a product (because it is simply a list, with its label), but it is not a *basic* product.

Data Object – A generic term for an object that is described by a description object. Data objects include both digital and non-digital objects.

Description Object – An object that describes another object. As appropriate, it will have structural and descriptive components. In PDS4 a 'description object' is a digital object, such as a string of bits with a predefined structure.

Digital Object – An object which consists of electronically stored (digital) data.

Identifier – A unique character string by which a product, object, or other entity may be identified and located. Identifiers can be global, in which case they are unique across all of PDS (and its federation partners). A local identifier must be unique within a label.

Label – The aggregation of one or more description objects such that the aggregation describes a single PDS product. In the PDS4 implementation, labels are constructed using XML (eXtensible Markup Language).

Logical Identifier (**LID**) – An identifier that identifies the set of all versions of a product.

Versioned Logical Identifier (LIDVID) – The concatenation of a logical identifier with a version identifier, providing a unique identifier for each version of product.

Metadata – Data about data. For example, a 'description object' contains information (metadata) about an 'object.'

Object – A single instance of a class defined in the PDS Information Model.

PDS Information Model – The set of rules governing the structure and content of PDS metadata. While the Information Model (IM) has been implemented in XML for PDS4, the model itself is implementation independent.

Product – One or more labeled objects (digital, non-digital, or both) grouped together and having a single PDS-unique identifier. In the PDS4 implementation, if a product consists of multiple objects, their descriptions are combined into a single XML label. Although it may be possible to locate individual objects within PDS (and to find specific bit strings within digital objects), PDS4 defines 'products' to be the smallest granular unit of addressable data within its complete holdings.

Registry – A data base that provides services for sharing content and metadata.

XML schema – The definition of an XML document, specifying required and optional XML elements, their order, and parent-child relationships.

2. OVERVIEW

2.1 Purpose and Scope

The purpose of this Software Interface Specification (SIS) is to provide users of the InSight Deployment Arm (IDA) archive with a detailed description of the data products and how they are generated, along with a description of the PDS4 archive bundle, the structure in which the data products, documentation, and supporting material are stored. The users for whom this document is intended are software developers of the programs used in generating the higher data products, and scientists who will analyze the data, including those associated with the InSight mission and those in the general planetary science community.

This SIS covers data products generated by IDA and the higher level products derived from them that are intended to be archived in the Planetary Data System (PDS).

2.2 Contents

This SIS describes how the IDA instrument acquires data, and how the data are processed, formatted, labeled, and uniquely identified. The document discusses standards used in generating the product and software that may be used to access the products. The data structure and organization are described in sufficient detail to enable a user understand the instrument, read the data and develop software to process these data.

2.3 Applicable Documents

[1] Planetary Data System Standards Reference, version 1.11.0, October 1, 2018.

[2] Data Provider's Handbook, Archiving Guide to the PDS4 Data Standards, version 1.11.0, October 1, 2018.

[3] PDS4 Data Dictionary Document, Abridged, version 1.11.0.0, September 23, 2018.

[4] PDS4 Information Model Specification, version 1.11.0.0, September 23, 2018.

[5] InSight Archive Generation, Validation and Transfer Plan, JPL D-75289, May 30 2014.

[6] InSight Flight-Ground Interface Document (FGICD), JPL D-75267.

[7] Trebi-Ollennu, A., et al., InSight Mars Lander Robotics Instrument Deployment System (2018). Space Sci Rev, 214:93.

[8] Soil mechanics IDA calibration Report (To be released after soil mechanics experiments have been performed)

[9] Shaw, A., et al., Phoenix Soil Physical Properties Investigation (2009). J Geophys Res, 114

[10] Smith, K., Technical Memorandum 20020233-B, Kinematic, Mass, and Stiffness Parameters of the Insight Robotic Arm

[11] Collins, C.L. and Robinson, M.L., Accuracy Analysis and Validation of the Mars Science Laboratory (MSL) Robotic Arm, in Proceedings of the ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Portland, Oregon, USA, 4-7 August 2013

[12] Spong, M.W. and Vidyasagar, M., Robot Dynamics and Control (1989) John Wiley and Sons

[13] Bonitz, R.G., et al., The Mars Surveyor ’01 rover and robotic arm, in IEEE Aerospace Conference, Proceedings, vol. 7 (2000), pp. 235-246

The PDS4 Documents [1] through [4] are subject to revision. The most recent versions may be found at pds.nasa.gov/pds4. The IDA PDS4 products specified in this SIS have been designed based on the versions current at the time, which are those listed above. Data products will be static and will not be changed if new versions of documents [1] to [4] become available.

2.4 Audience

This document serves both as a Data Product SIS and an Archive SIS. It describes the format and content of IDA data products in detail, and the structure and content of the archive in which the data products, documentation, and supporting material are stored. This SIS is intended to be used both by the instrument team in generating the archive and by data users wishing to understand the format and content of the archive. Typically, these individuals would include scientists, data analysts, and software engineers.

2.5 InSight Mission

InSight was launched on May 5, 2018 and placed a single geophysical lander on Mars on November 26, 2018, to study its deep interior. The Surface Phase consists of Deployment and Penetration, and Science Monitoring. It ends after one Mars year plus 40 sols.

The science payload includes two instruments: the Seismic Experiment for Interior Structure (SEIS) and the Heat-Flow and Physical Properties Package (HP^3). In addition, the Rotation and Interior Structure Experiment (RISE) uses the spacecraft X-band communication system to provide precise measurements of planetary rotation. SEIS and HP^3 are placed on the surface with an Instrument Deployment System (IDS) comprising an Instrument Deployment Arm (IDA), Instrument Deployment Camera (IDC), and Instrument Context Camera (ICC). There are also several supporting instruments. The Auxiliary Payload Sensor Subsystem (APSS) includes the pressure sensor, the magnetometer, and Temperature and Wind for InSight (TWINS) sensors and collects environmental data in support of SEIS. These data are used by SEIS to reduce and analyze their data. The radiometer (RAD) is used by the HP^3 team to measure surface temperature and thermal properties to support their data analysis. This document describes the IDA and the associated data products.

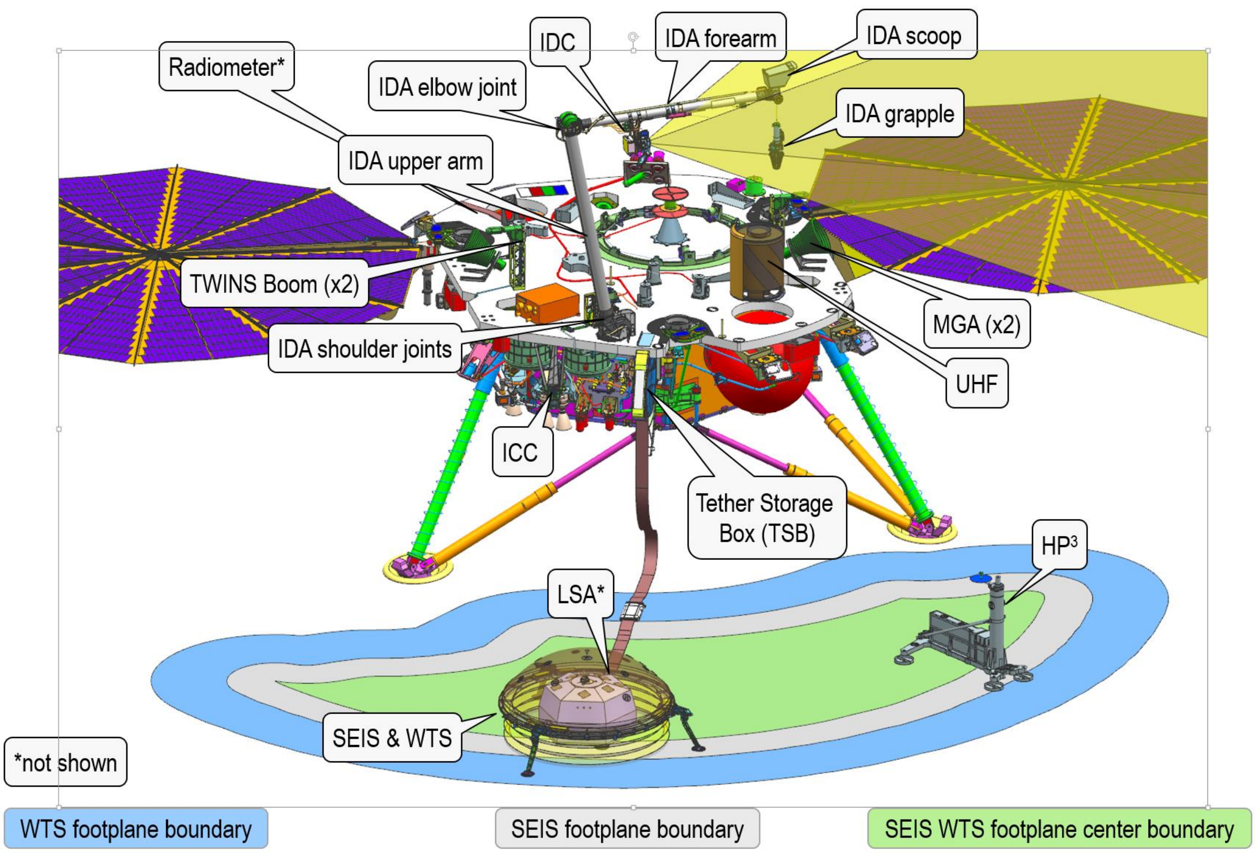


Figure 1 - InSight lander with instrument and IDA elements labeled

2.6 IDA Description

The purpose of the IDA is threefold: firstly, to deploy the SEIS, the SEIS Wind and Temperature shield (WTS), and HP^3 instruments from the InSight lander deck to the Martian surface; secondly, to point the IDC for imaging of the Martian terrain surrounding the InSight lander; thirdly, to perform soil mechanics experiments.

For more information about the IDA, refer to Trebi-Ollennu, et al. 2018 [7].

2.6.1 IDA overview

The IDA is the flight Robotic Arm (RA) from the original Mars `01 lander. The various components of the IDA are illustrated in Figure 1 and Figure 2. The IDA is a four degree-of-freedom (DOF) back-hoe design manipulator that provides the following motion: yaw (shoulder azimuth, joint 1), and three pitch joints (shoulder elevation, elbow, and wrist, joints 2 through 4, respectively). Each of the IDA joints consists of a brushed DC motor with two-stage planetary gears and a harmonic drive at the output (except the wrist, which has a bevel gear at the output of the planetary gears). During normal operations, the IDA actuators are capable of generating 35, 120, 65, and 10.5 Newton-meters of torque at the joint output for joints 1 through 4, respectively. Each joint is equipped with a temperature sensor, a heater, and two position sensors: encoders on the joint input motor shaft and potentiometers at the joint output load shaft.

The IDA end-effector consists of a grapple, a scoop, and the IDC.

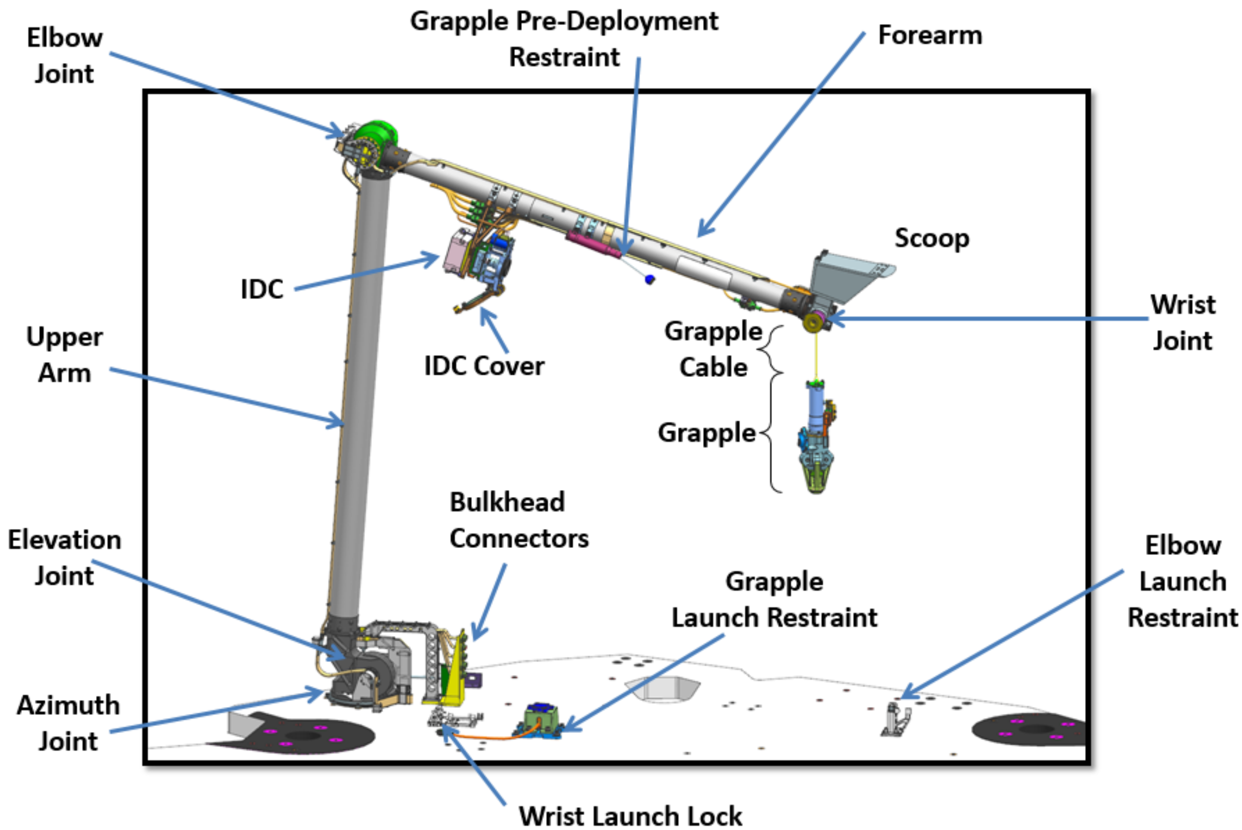


Figure 2 - IDA on the lander deck

2.6.1.1 IDA End-Effectors

2.6.1.1.1 IDA End-Effector Grapple

The grapple is a five finger "claw" and hangs by an umbilical cable at the IDA end-effector. Using the grapple, the IDA can lift and deploy a 9-kg payload on Mars (33N) at 1.65 m distance.

The grapple is stowed against the IDA forearm such that it does not obstruct the IDC Field of View (FOV). However, during deployment, the grapple is unstowed and hangs in the IDC FOV such that the IDC images can capture the opening of the grapple fingers and engagement of spherical cap grapple hooks on the payload. The IDA can position the grapple to capture the payload's spherical cap grapple hook, lift, and place SEIS, WTS, and HP[3] on the Martian surface. The grapple can be re-stowed against the IDA after instrument deployment.

The grapple fingers are opened by a single High Input Paraffin (HOP) actuator that slowly heats up, melts the wax that pushes a rod out to open the fingers. As the grapple HOP cools down in the ambient temperature, the grapple fingers slowly close passively without any actuation.

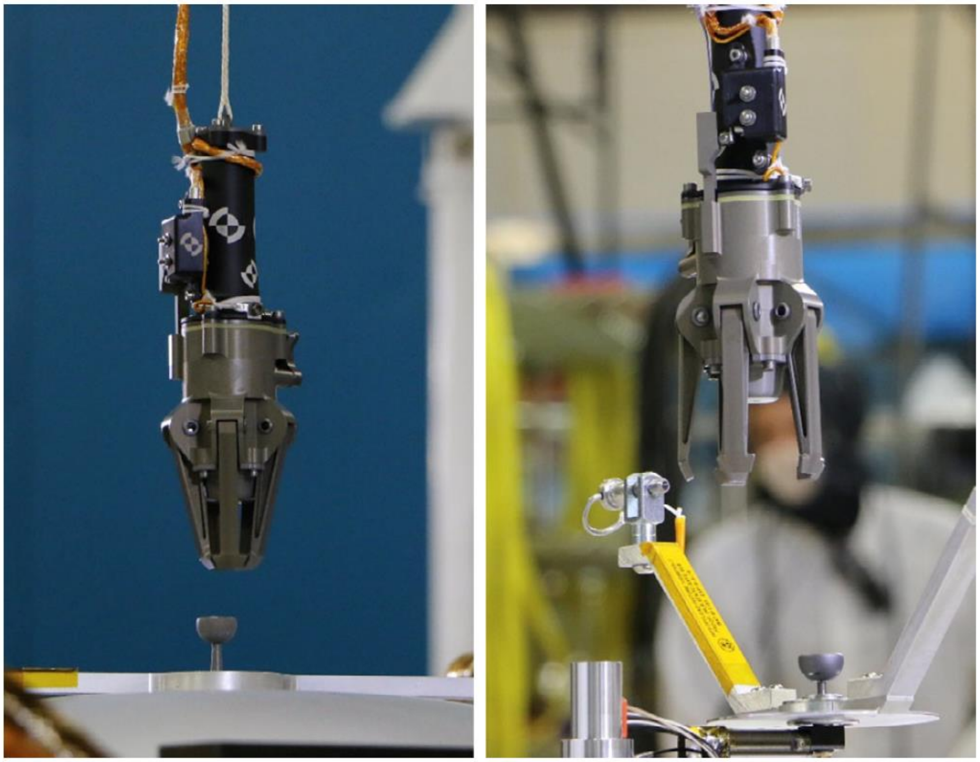


Figure 3 - End-effector grapple with fingers closed (left) and fingers opened (right)

2.6.1.1.2 IDA End-Effector Scoop

The scoop consists of a single chamber with a front blade and a secondary blade on the bottom side. The scoop will enable soil mechanics experiments for inferring mechanical properties of the Martian soil at the landing site. The scoop's applied force is configuration dependent, but the average force is typically about 80N. The scoop is not required for nominal instrument deployment operations.

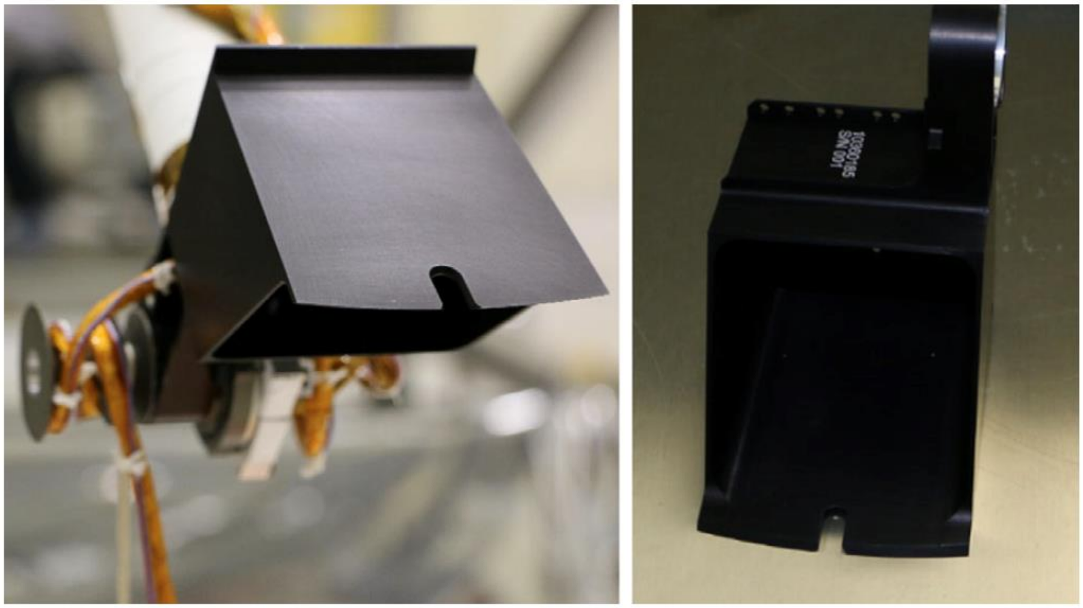


Figure 4 - End-effector scoop

2.6.1.1.3 IDA End-Effector IDC

The IDC is a forearm mounted camera (closer to the elbow joint) facing the IDA end-effector. The IDA is used to point the IDC to take images of the surface, lander (selfie), lander elements, 360° panorama of the landing site, and other geological features at the landing site. The IDC allows visual confirmation of deployment steps, acquisition of the stereo image pairs used to create a 3D terrain map of the workspace, and provides engineering images of the solar arrays, payload deck, and instruments.

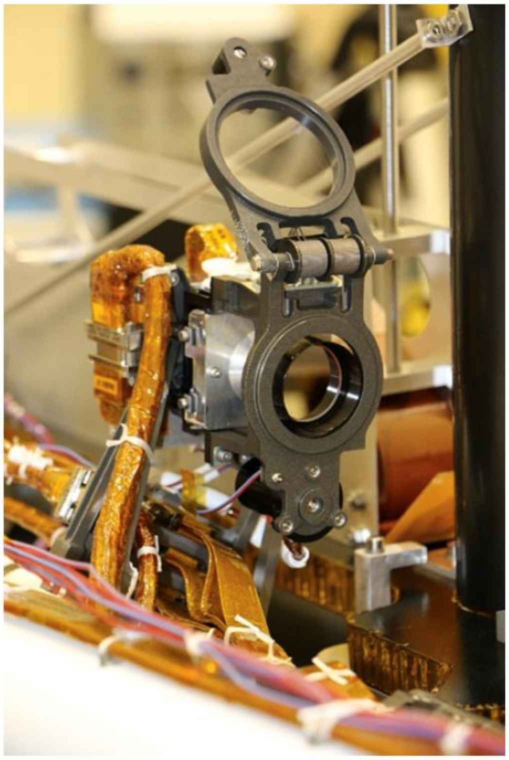


Figure 5 - End-effector IDC with dust cover opened

2.6.1.2 IDA Motor Controller

The IDA Motor Controller (MC) consists of two printed-circuit boards located in the lower Payload Electronics Box (PEB) and provides power conditioning, motor voltage control and drivers, grapple heater drivers, joint encoder counting, and Analog-to-Digital (A/D) conversion of potentiometer voltages, temperature sensor voltages, motor currents, and heater current. The PEB provides the interface to the Lander Command and Data Handling (C&DH) computer via the InSight Payload Interface Card (IPIC) board. Firmware residing on the IDA MC microprocessor provides for low-level motor command execution to move the joints to the specified positions, grapple heater command execution, A/D calibration, and sensor monitoring.

2.6.1.3 IDA Flight Software

The IDA Flight Software (FSW) provides both control of, and visibility into, the IDA hardware. It runs on board the C&DH computer and communicates with the PEB. IDA FSW provides to following specific capabilities:

- Interface with external entities, including other spacecraft FSW components and the IDA PEB
- Expansion of high-level IDA commands from the command sequencer into low-level IDA actions
- Motion control of the IDA
- Control of the grapple
- Fault sensing, recovery, and safing
- Collision prevention between the IDA, lander, and science instruments
- Visibility of the IDA state in telemetry

The IDA FSW responds to sequences of IDA commands which specify the desired movement in terms of the goal joint angles, a Cartesian position and approach angle for the specified tool, or a direction and length of time to move certain motors. IDA FSW breaks motion commands down into a sequence of via points describing intermediate joint positions to achieve the desired motion. The IDA FSW provides the command decomposition into via points, trajectory generation for each via point, and fault monitoring.

The IDA kinematics describe the geometrical relationships among the IDA elements. The forward kinematics maps IDA joint to the pose of the selected tool in Cartesian space. The inverse kinematics maps the pose of the tool to the corresponding joint angles. IDA FSW computes the forward and inverse kinematics to enable and determine IDA placement.

2.6.2 IDA operations

2.6.2.1 Workspace Imaging and Terrain Mosaic

Prior to deployment, an IDC stereo mosaic of the workspace is acquired to create a Digital Elevation Model (DEM) that provides information of the workspace terrain in 3-D coordinates. In order to minimize stereo baseline error, IDC stereo pairs are acquired by moving one IDA joint only – the shoulder joint (azimuth) – while keeping all the other joints constant. IDC workspace imaging is done in several tiers, starting with an inner tier close to the base of the lander and moving progressively outward. Only the IDA azimuth joint angle is changed within a tier. The IDA elbow joint angle is changed to move from one tier to the next. The IDC image data products are available online and are released via the PDS Cartography and Imaging Sciences Node (https://pds-imaging.jpl.nasa.gov/volumes/insight.html).

2.6.2.2 Payload Deployment Steps

Each payload deployment (lift from the Lander deck to placement on surface) consists of four parts (Part 1, Part 2, Part 3, and Part 4).

- Part 1 of each payload deployment entails moving the unstowed grapple to the corresponding payload teach point. The payload teach points are 5 cm above the grapple hook position of the payload as stowed on the Lander deck.
- Part 2 consists of opening the grapple fingers and moving the fully open grapple fingers 4 cm down to capture the payload.
- Part 3 entails lifting the captured payload and placing it on the surface of Mars.
- Part 4 is called payload release, and it consists of opening the grapple and moving the IDA up and away from the payload grapple hook.

2.6.2.3 Soil mechanics experiments

For the soil mechanics experiments, the IDA will be used as a science instrument to investigate the physical properties of the Martian surface at the InSight landing site. The IDA provides guarded motion capability, a single command that allows the IDA to be commanded to move to a position until contact is made. Guarded move command capability will enable the IDA soil mechanics experiments, that consist of indentation tests. Analysis that used similar data products to infer soil physical properties from the Robotic Arm activities during the Phoenix mission can be found in [9].

2.6.3 IDA calibration

On the ground, two thermal characterization tests were performed on the IDS subsystem in a thirteen-foot Sensor Chamber at the Raytheon El Segundo Integrated Test Laboratory (ITL), California. During the test, the IDA heaters were characterized and IDA functional qualification was successfully performed at proto-flight operational temperature. In addition, IDA stop-and-hold torques were characterized at various temperatures.

3. IDA DATA PRODUCTS

3.1 Data Product Overview

IDA data products consist of raw observations and calibrated data. When the IDA is commanded and operational, Science/Engineering Low Priority (SciLo) and Science/Engineering High Priority (SciHi) data products are provided. Additionally, IDA status, IDA history, and IDA parameters are downlinked and included in this archive. Supplemental RSVP Replay Videos generated using the Rover Sequencing and Visualization Program (RSVP) are included in the PDS archive when the soil mechanics experiments are performed.

Table 4 shows a summary of IDA data products. See Table 5 for processing level definitions. Details about each product can be found in Section 5.

Table 4 - IDA data products overview

Product	Processing level	Storage format
IDA status	Ancillary	ASCII table
IDA history	Ancillary	ASCII table
IDA parameters	Ancillary	ASCII table
Science/Engineering Low Priority (SciLo) data	Raw/Calibrated	ASCII text tables
Science/Engineering High Priority (SciHi) data	Raw/Calibrated	ASCII text tables

3.2 Data Processing

This section describes the processing of IDA data products, their structure, organization, and labeling.

3.2.1 Data Processing Level

Data processing levels mentioned in this SIS refer to the PDS4 processing level described in Table 5. The lowest processing level archived in PDS is “raw” as described in the table. PDS does not archive telemetry.

Table 5 - Data Processing Level Definitions

PDS4 processing level	PDS4 processing level description
Raw	Original data from an experiment. If compression, reformatting, packetization, or other translation has been applied to facilitate data transmission or storage, those processes are reversed so that the archived data are in a PDS approved archive format. Often called EDRs (Experimental Data Records).
Partially Processed	Data that have been processed beyond the raw stage but which have not yet reached calibrated status. These and more highly processed products are often called RDRs (Reduced Data Records).
Calibrated	Data converted to physical units, which makes values independent of the experiment.
Derived	Results that have been distilled from one or more calibrated data products (for example, maps, gravity or magnetic fields, or ring particle size distributions). Supplementary data, such as calibration tables or tables of viewing geometry, used to interpret observational data should also be classified as ‘derived’ data if not easily matched to one of the other three categories.

IDA data product described in this SIS are a hybrid of raw and calibrated processing levels.

3.2.2 Data Product Generation

When the IDA is commanded and operational, it generates blocks of data in regular intervals which are stored on-board. Upon further commanding, the stored data is processed on-board by the downlink processor and sent down to Earth in telemetry packets. See Appendix C for the structure of the IDA primary data format.

On the ground, the Multimission Instrument Processing Laboratory (MIPL) subsystem is responsible for processing these packets and generating instrument products which are collections of time based, sorted packets. The IDA generates and downlinks a number of different data types, known as Application Identifier (APID). Data for each APID is stored as a separate product. Packets with different APIDs contain different type of data, with different structure.

The MIPL subsystem collects and processes six different telemetry data for the IDA. They are listed in Table 6.

Table 6 - IDA telemetry data processed by MIPL

APID	Description
84	Collision database

85	IDA history, from onboard ring buffer of 300 status products (APID 89)
86	IDA internal parameters and configuration
87	Science/Engineering data, high downlink priority
88	Science/Engineering data, low downlink priority
89	IDA status

MIPL generates four types of data products for IDA:

- Collision data product for APID 84
- History and status products for APIDs 85 and 89
- Parameter and configuration for APID 86
- Science and engineering for APIDs 87 and 88

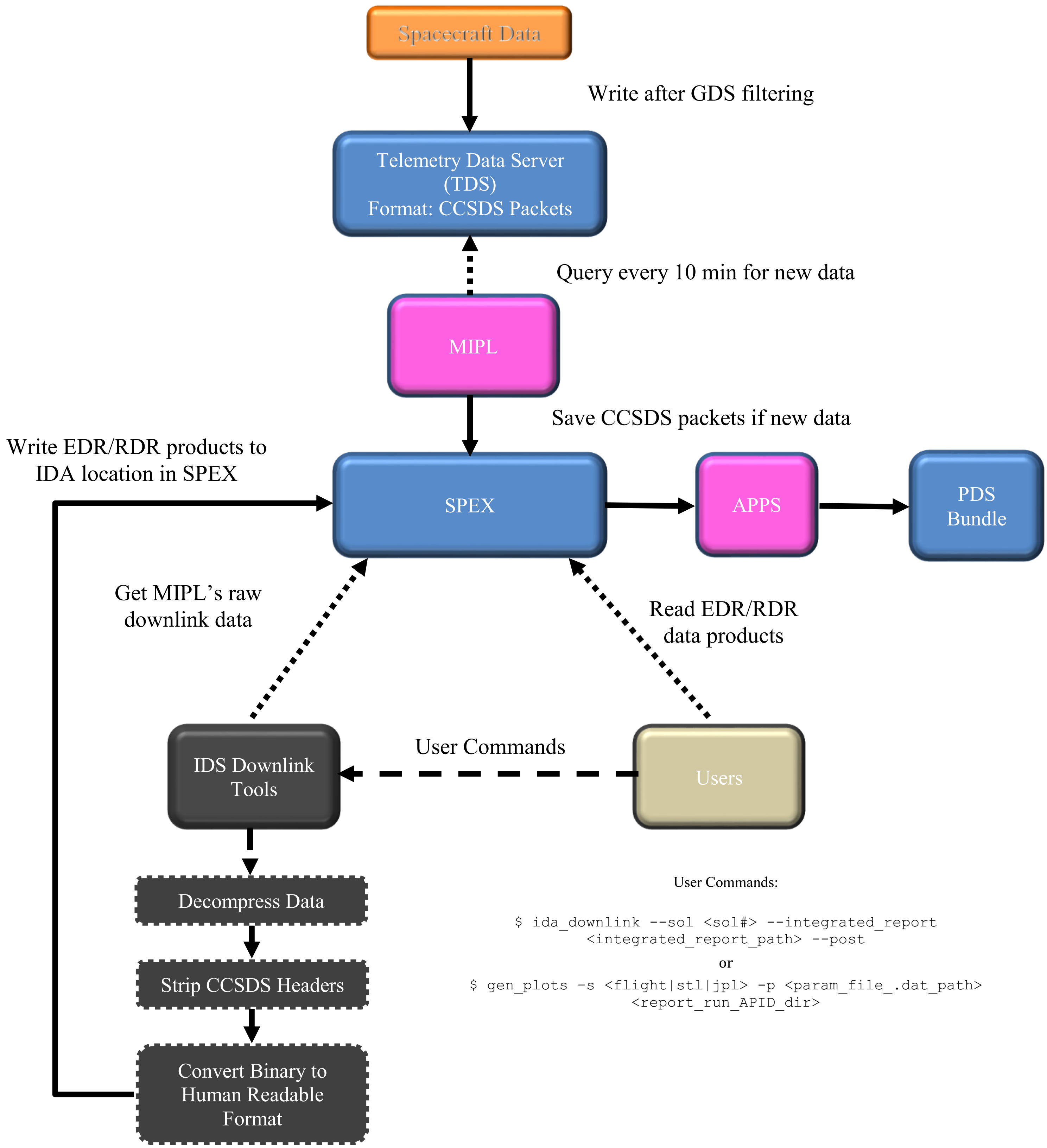
The format and structure for all the products is similar. The files contain the original CCSDS (Consultative Committee for Space Data Systems) packets for a given APID from a given downlink, DSN pass. It is possible to have a gap, or a jump, in the SCLK, but the packets will always be sorted by SLCK. If there are packets that are missing for a given downlink, they will be put into the file/product that is generated for the future pass that contains that data.

3.2.3 Data Flow

This section describes only those portions of the InSight data flow that are directly connected to IDA archiving. A full description of InSight data flow is provided in the InSight Archive Generation, Validation, and Transfer Plan [5].

IDA data are downlinked by InSight Project Operations at JPL. The JPL Advanced Multimission Operations System (AMMOS) Multimission Instrument Processing Laboratory (MIPL) delivers raw CCSDS telemetry packets to the IDA team at JPL. The IDA team extracts raw data products from the packets. The MIPL team generates PDS4 labels for all the IDA data products, assembles the data and documentation into IDA archive bundles, and delivers the bundles to the PDS Geosciences Node. Deliveries take place according to the release schedule agreed upon by the InSight Project and PDS and specified in the InSight Archive Plan. The Geosciences Node validates the bundles for PDS4 compliance and for compliance with this SIS document, and makes them available to the public online.

Spacecraft IDA Data Flowchart



3.3 Standards Used in Generating Data Products

The IDS data products comply with the Planetary Data System PDS4 standard for science data archives, as specified in the PDS Standards Reference [1], the Planetary Science Data Dictionary Document [3], and the PDS4 Information Model Specification [4]. These standards documents are revised approximately every six months. The version that is current at the time of the IDA peer review is used for the archive. See Section 4, for a description of the PDS Label and the specific conventions.

3.3.1 Time Standards

PDS labels for InSight data products allow the use of various time-related attributes. The following are used in labels for IDA products.

Time value	Label attribute*	Formation rule
UTC time of observation	**start_date_time**, **stop_date_time**	yyyymmddThh:mm:ss.ffffffZ
Sol (Mars day) of observation	**sol_number**, or the pair **start_sol_number**, **stop_sol_number**	A non-negative integer (landing day is Sol 0)
Spacecraft clock count (SCLK)	**spacecraft_clock_start count**, **spacecraft_clock_stop_count**	[p/]dddddddddd[-fffff] where p = partition number, dddddddddd = whole seconds, fffff = fractional seconds as 1/65536 ticks
	spacecraft_clock_count_partition must be used if SCLK values do not include a partition number	A positive integer
*Attributes start_date_time and stop_date_time are defined in the PDS4 core dictionary. The others are defined in the InSight Mission Dictionary.		

See an example of IDA label in Appendix E.

3.3.2 Coordinate Systems

This section describes the primary coordinate systems defined for surface operations, which are listed in Table 7 and illustrated in Figure 6 below.

Table 7 - Coordinate Frames Used for Surface Operations

Name	Label Keyword	Origin	Orientation
Payload Frame (P Frame) (IDA Frame)	"PAYLOAD_FRAME"	Attached to Lander at intersection of IDA joint 1 rotation axis and the top surface of the lander deck	Fixed offset frame rotated 180 deg about +Z axis relative to Lander frame: • +Z axis is normal to deck surface and points from that surface downward.

			• +X axis is perpendicular to Z axis and points towards IDA side of deck. • +Y completes the right-handed frame.
Lander Frame (L Frame) NOTE: Not used in Surface Operations	does not appear in label	Centered on launch vehicle separation plane 956.056 mm above Lander deck. Lander Mechanical frame origin relative to the Payload Frame = [-775.084, -283.360, -956.056]	Aligned with Lander: • +Z axis is normal to deck surface and points from that surface downward. • +X axis is perpendicular to Z axis, parallel to solar array yoke and points toward deck side opposite to IDA. • +Y completes the right-handed frame.
IDC Frame (S_{IDC} Frame)	does not appear in label	Attached to Camera	Aligned with camera pointing
Site (S_N Frame) (Surface Frame)	does not appear in label	Attached to Surface	Aligned with Lander: • +Z axis points downward to Nadir (gravity vector). • +X axis is perpendicular to Z axis and points towards North. • +Y completes the right-handed frame and points East.
Local Level (L_L Frame)	does not appear in label	Attached to Lander (coincident with Payload Frame)	Fixed offset frame relative to Site frame: • +Z axis points downward to Nadir (gravity vector). • +X axis is perpendicular to Z axis and points towards North. • +Y completes the right-handed frame and points East.
SEIS ASM (Frame)		Centered at SEIS grapple hook	The main frame used by the SEIS team. It is fixed to SEIS: • +Z axis is upwards • +X axis is oriented from the sphere center to the dampers located at 36 degrees of the sphere valve • +Y axis completes the right-handed coordinate frame
SEIS Grapple Hook Frame		Centered at SEIS grapple hook	Defined along the tether: • +Z axis is co-aligned with the IDA Z axis and is also fixed to SEIS

			• +X axis is along the tether on the deck and fixed to SEIS so the X axis always faces the opposite direction of the LSA • +Y axis completes the right-handed coordinate frame

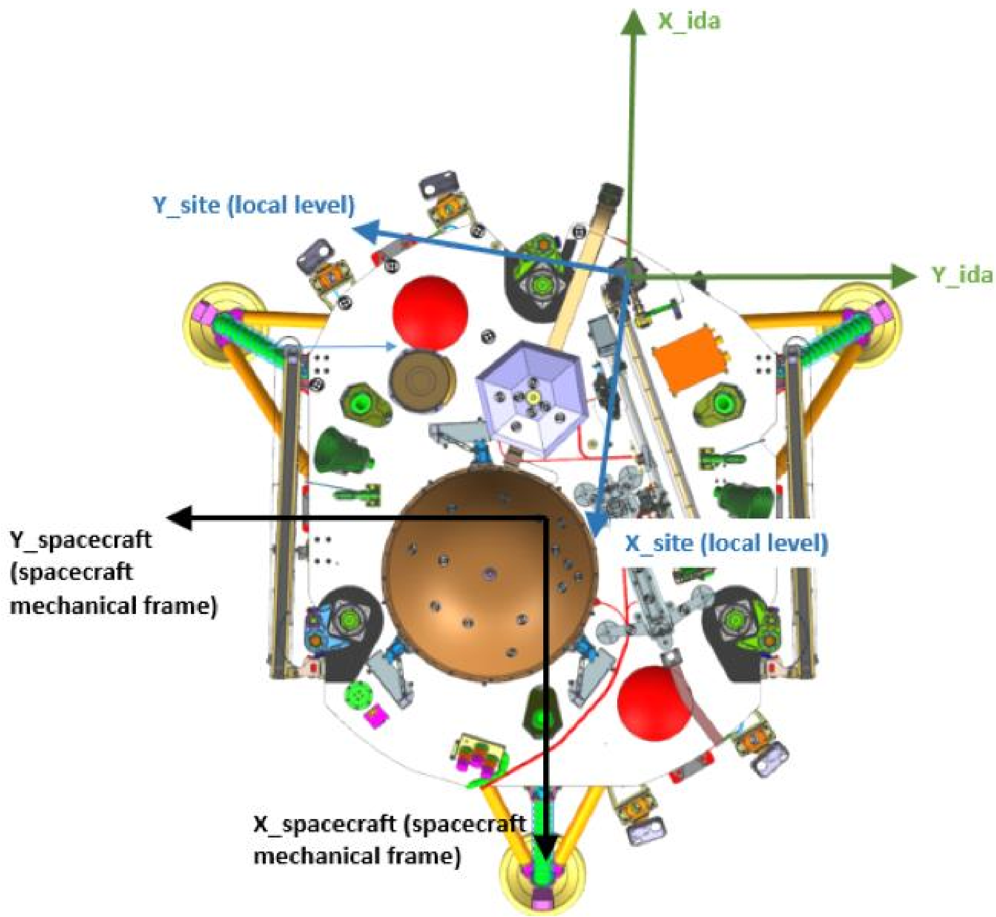


Figure 6 - IDA Frame and Site (local level) Frame

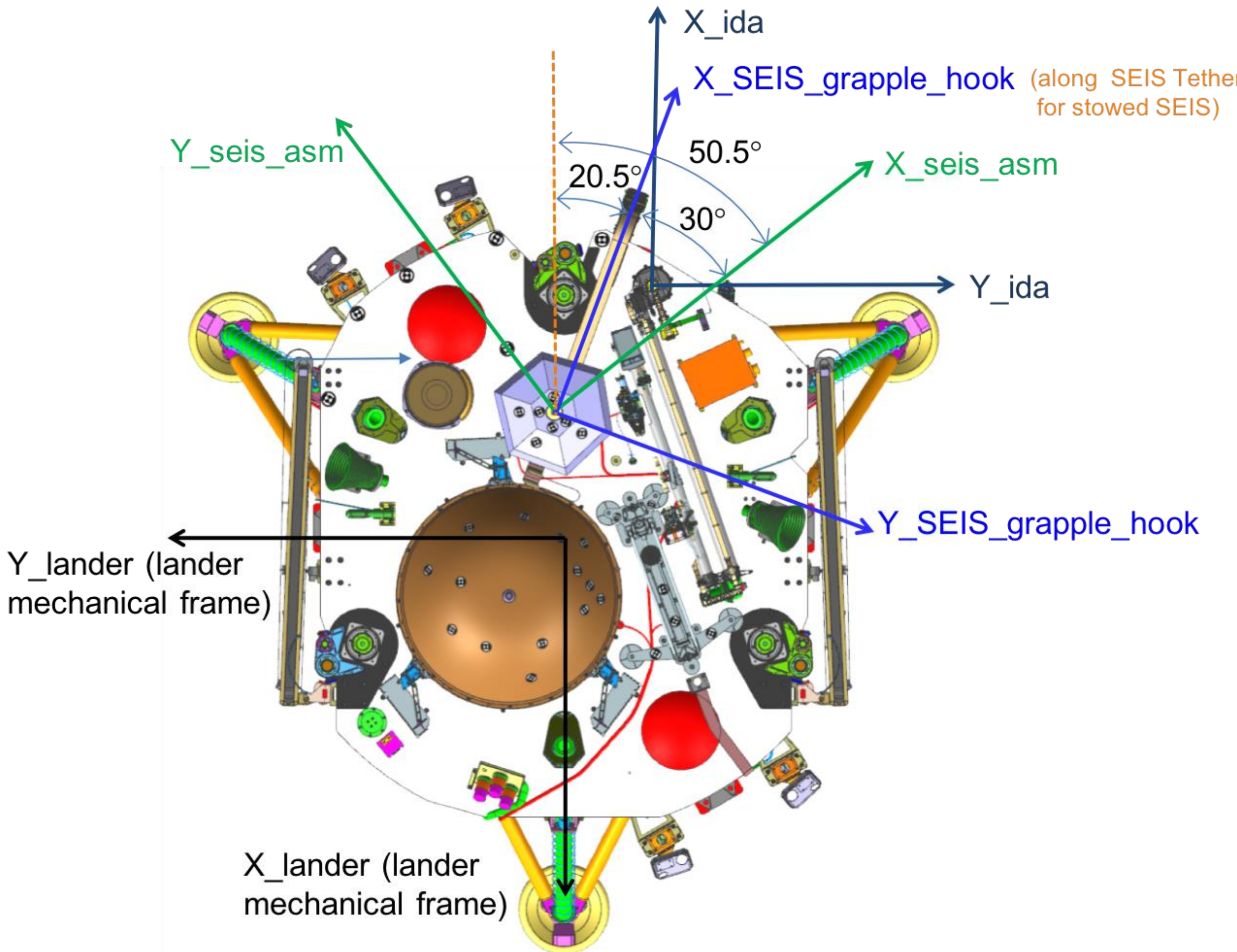


Figure 7 - IDA, SEIS ASM, and SEIS Grapple Hook Coordinate Frames

3.3.3 Data Storage Conventions

IDA products are stored as ASCII fixed-width tables.

3.4 Applicable Software

IDA data in ASCII tables may be read using many text editors and spreadsheet programs. No special software is required for use with IDA data.

3.5 Backups and Duplicates

The Geosciences Node keeps online copies of each archive product. One copy is the primary online archive copy. Another is a backup copy. Once the archive products are fully validated and approved for inclusion in the archive, another copy of the archive is sent to the National Space Science Data Center (NSSDC) for long-term preservation in a NASA-approved deep-storage facility. The Geosciences Node may maintain additional copies of the archive products, either on- or off-site as deemed necessary according to the Node's backup and disaster recovery plan.

4. IDA ARCHIVE ORGANIZATION, IDENTIFIERS AND NAMING CONVENTIONS

This section describes the basic organization of the IDA data archive under the PDS4 Information Model (IM) (Applicable documents [1] and [3]), including the naming conventions used for the bundle, collection, and product unique identifiers. The formation of logical identifiers is described in Section 4.1. Bundles, collections and products are defined and given identifiers in Section 4.2. In short, a group of related products forms a collection, and a group of related collections forms a bundle.

4.1 Logical Identifiers

Every product in PDS is assigned a Logical Identifier (LID) that allows it to be uniquely identified across the system. Each product also has a Version Identifier (VID) that allows different versions of a specific product to be referenced uniquely. A product’s LID and VID are defined as separate attributes in the product label. For convenience they may be combined in a single string called a LIDVID, with two colons between the LID and the VID. If a particular version of a product is desired, the LIDVID should be used; otherwise the LID alone should be used with the understanding that it refers to the latest version of the product.

LIDs and VIDs are assigned by PDS and are formed according to the conventions described in the following sections. More information on LIDs and VIDs may be found in Section 6d of the PDS Standards Reference [1] and in Chapter 5 of the Data Providers' Handbook [2].

4.1.1 LID Formation

LIDs take the form of a Uniform Resource Name (URN). LIDs are restricted to ASCII lower case letters, digits, dash, underscore, and period. Colons are also used, but only to separate prescribed components of the LID. Within one of these prescribed components dash, underscore, or period are used as separators. LIDs are limited in length to 255 characters.

InSight IDA LIDs are formed according to the following conventions:

- Bundle LIDs are formed by appending a bundle-specific ID to the PDS base ID:

  urn:nasa:pds:<bundle ID>

  Example: urn:nasa:pds:insight_ida

  The bundle ID must be unique across all bundles archived with the PDS.

- Collection LIDs are formed by appending a collection-specific ID to the collection’s parent bundle LID:

  urn:nasa:pds:<bundle ID>:<collection ID>

  Example: urn:nasa:pds:insight_ida:data_ancillary

  The collection ID must be unique across the bundle. Collection IDs correspond to the collection type (e.g. “browse”, “data”, “document”, etc.). Additional descriptive information may be appended to the collection type (e.g. “data-raw”, “data-calibrated”, etc.) to ensure that multiple collections of the same type within a single bundle have unique LIDs.

- Basic product LIDs are formed by appending a product-specific ID to the product's collection LID:

  urn:nasa:pds:<bundle ID>:<collection ID>:<product ID>

  Example:urn:nasa:pds:insight_ida:data_ancillary:a0089_0062_602047157_602052923_190129195008_006_arm_float

The product ID must be unique across the collection. For IDA data products, the product LID is the same as the lower case data file name without the extension.

4.1.2 VID Formation

Product Version IDs consist of major and minor components separated by a "." (M.n). Both components of the VID are integer values. The major component is initialized to a value of "1", and the minor component is initialized to a value of "0". The minor component resets to "0" when the major component is incremented. The PDS Standards Reference [1] rules for incrementing major and minor components.

4.1.3 File Naming Conventions

All IDA data products are named according to the following naming conventions:

A	APID	_	Start Sol	Start Sclk	_	End Sclk	_	yy	mm	dd	hh	mm	ss	_	Product Identifier
A	0089	_	0062	602047157	_	602052923	_	19	01	29	19	50	08	_	006_arm_float.txt

Where:

- A indicates that the product is an IDA product
- [APID] identifier specifies the APID number (Table 6) as a four digit number
- [Start Sol] identifier specifies the Martian sol (counted after InSight landing) when the data started to be collected by the instrument as a four digit number
- [Start SCLK] identifier specifies the SCLK of the earliest packet as a nine digit number
- [End SCLK] identifier specifies the SCLK of the latest packet as a nine digit number
- [yy][mm][dd][hh][mm][ss] is the time tag identifier and corresponds, respectively, to the numerical value of year, month, day, hours, minutes and seconds UTC when the data started to be collected by the IDA
- [Product] identifier specifies the data product and its file extension. It can be one of the following:

Table 8 - File Naming Convention

Data Product	[Product] identifier
IDA status	006_arm_float.txt
IDA history	000_system_float.txt
IDA parameters	012.txt
Engineering/Science Low Priority (SciLo)	eu.csv
Engineering/Science High Priority (SciHi)	eu.csv

An example of data product name is:

A0089_0062_602047157_602052923_190129195008_006_arm_float.txt

This file contains IDA status data collected starting on sol 62 of the mission, with a starting SCLK time of 602047157 and an end SCLK time of 602052923, starting data acquisition on the 29th of January 2019 at 19:50:08 UTC.

4.2 IDA Bundles

The highest level of organization for a PDS archive is the bundle. A bundle is a set of one or more related collections which may be of different types. A collection is a set of one or more related basic products which are all of the same type. Bundles and collections are logical structures, not necessarily tied to any physical directory structure or organization.

The complete InSight IDA archive is organized into one bundle (Table 9).

Table 9 - IDA Bundle

Bundle Logical Identifier	PDS4 Processing Level	Description
urn:nasa:pds:insight_ida	Raw, Calibrated	The **insight_ida** bundle contains collections for IDA raw data, and documentation, including time-lapse videos of data acquisition.

In addition to the IDA data bundle, the InSight archives have a Document Bundle that contains documentation for all InSight data bundles, including IDA documents. The contents of the Document Bundle are described in Section 4.5.

4.3 IDA Collections

The IDA Bundle contains the collections listed in Table 10.

Table 10 - Collections in the IDA Bundle

Collection Logical Identifier	Collection Type	Description
urn:nasa:pds:insight_ida:data_ancillary	Ancillary data	IDA status, history, and parameters
urn:nasa:pds:insight_ida:data_raw_calibrated	Data	Hybrid raw and calibrated IDA data products
urn:nasa:pds:insight_ida:document_video	Document	Time-lapse videos of data acquisition
urn:nasa:pds:insight_documents:document_ida	Document	Documentation, including the Software Interface Specification (SIS) and the Soil mechanics calibration report

4.4 IDA Products

An IDA science data product consists of one digital object in one file, accompanied by a PDS label file. The PDS label provides identification and description information for the data file. As discussed above, the PDS label includes a Logical Identifier (LID) by which the product is uniquely identified throughout all PDS archives. Under the PDS4 standard, labels are XML-formatted ASCII files.

In addition to data products, the IDA archive contains document products. These also have PDS labels.

Finally, the collections and bundles themselves are considered "products" in PDS, and have their own PDS labels.

4.5 InSight Document Bundle and Collections

Documents are also considered products in PDS, and have LIDs, VIDs and PDS4 labels just as data products do. The InSight archives include an InSight Document Bundle, which consists of collections of documents relevant to the mission itself and all the science experiments. The IDA Team is responsible for the IDA document collection in this bundle.

Table 11 - Collections in the InSight Document Bundle

Collection Logical Identifier	Description
urn:nasa:pds:insight_documents:document_mission	InSight mission, spacecraft and lander descriptions
urn:nasa:pds:insight_documents:document_apss	APSS SIS, instrument description, and other relevant documents
urn:nasa:pds:insight_documents:document_camera	Camera SIS, instrument description, and other relevant documents
urn:nasa:pds:insight_documents:document_hp3rad	HP^3/RAD SIS, instrument description, and other relevant documents
urn:nasa:pds:insight_documents:document_ida	IDA SIS (this document), instrument description, and other relevant documents
urn:nasa:pds:insight_documents:document_mag	MAG SIS, instrument description, and other relevant documents
urn:nasa:pds:insight_documents:document_rise	RISE SIS, instrument description, and other relevant documents
urn:nasa:pds:insight_documents:document_seis	SEIS SIS, instrument description, and other relevant documents
urn:nasa:pds:insight_documents:document_spice	SPICE relevant documents

In PDS4 a collection may belong to more than one bundle. It is a primary member of one bundle – the one on which its LID is based – and a secondary member of other bundles. The collection **urn:nasa:pds:insight_documents:document_ida** is a primary member of the InSight Document

Bundle and a secondary member of the IDA Bundle. The actual files that comprise the collection are found in the bundle where it is a primary member. They do not have to be duplicated in the other bundles, although they may be.

5. IDA ARCHIVE PRODUCT FORMATS

Data that compose the IDA data archive are formatted in accordance with PDS specifications (Applicable Documents [1], [2], and [3]). This section provides details on the formats used for each of the products included in the archive.

5.1 Data Product Formats

This section describes the format and record structure of each of the data file types. IDA data products will be stored as ASCII text tables and ancillary ASCII files.

5.1.1 IDA Status

IDA arm status files contain reported values from the PEB. The status data is the following:

```
ra_status.ace.calib_OK     =  1
ra_status.ace.bus30v_err   =  0
ra_status.ace.parity_err   =  0
ra_status.ace.msg_err      =  0
ra_status.ace.load_box     =  0
ra_status.ace.self_test    =  0

ra_status.mtr.pwr_on       = [  0     0     0     0    ]
ra_status.mtr.rotation     = [  0     0     1     0    ]
ra_status.mtr.over_current = [  0     0     0     0    ]
ra_status.mtr.brake        = [  0     0     0     0    ]
ra_status.mtr.current      = [  0.000  0.001  0.000  0.001 ]

ra_status.jnt.enc          = [ -0.100  -0.320   1.010   0.134  ]
ra_status.jnt.pot          = [ -0.100  -0.321   1.009   0.134  ]
ra_status.jnt.temp         = [  26.512  26.656  28.098  28.538 ]

ra_status.ref              = [  4.502 -4.498 ]
ra_status.htr_crnt         =  0.001
ra_status.agnd             =  0.002

ra_status.time             =  594733695.9
```

- ra_status.ace.calib_OK – Boolean whether PEB calibration is nominal.
- ra_status.ace.bus30v_err – Boolean whether PEB 30v Bus is in an error state.
- ra_status.ace.parity_err – Boolean whether PEB parity bit is in an incorrect state.
- ra_status.ace.msg_err – Boolean whether PEB command had format error.
- ra_status.ace.load_box – Boolean whether PEB load box is present. Note that this value should always be 0 for flight.
- ra_status.ace.self_test – Boolean whether PEB is in a self-test state.
- ra_status.mtr.power_on – joint 1, joint 2, joint 3, joint 4 boolean if motor is on.
- ra_status.mtr.rotation – joint 1, joint 2, joint 3, joint 4 direction of rotation.
    - 0 – positive
    - 1 – negative

- ra_status.mtr.over_current – joint 1, joint 2, joint 3, joint 4 boolean whether an over-current has been detected.
- ra_status.mtr.brake – joint 1, joint 2, joint 3, joint 4 boolean whether motor brake is on
    - 0 – off
    - 1 – on
- ra_status.mtr.current – joint 1, joint 2, joint 3, joint 4 current in amps
- ra_status.ref – vref1, vref2 reference voltage in volts
- ra_status.htr_crnt – Sum of heater currents in amps
- ra_status.agnd – Analog ground value in volts
- ra_status.time – The last time (SCLK) when the above data was acquired from the PEB

IDA system status files contain reported values from IDA FSW. During a motion command, the status values are recorded at a constant rate. During a non-motion command, the status values are recorded if there is a change from the last recorded values. The status data is the following:

```
sys_status.curr_limit              = [ 172 325 273 190 ]
sys_status.mtr_volt                = [ 2 1 2 1 ]
sys_status.brake                   = [ 0 0 0 0 ]
sys_status.use_pots                = [ 0 0 0 0 ]
sys_status.htr_on                  = [ 0 0 0 0 ]
sys_status.tool                    = 5  RA_GRAPPLE
sys_status.cool_down               = 0
sys_status.fault_chk_mask          = [ F00F0F 39FF7E7F ]
sys_status.fault_status            = [ 0 0 ]
sys_status.act_cmd                 = 23  RA_SEND_DATA
sys_status.act_args                = 22 0
sys_status.task_cmd                = 23  RA_SEND_DATA
sys_status.task_args               = 22 0
sys_status.target.q                = [  0.026 -0.389  1.378  0.125 ]
sys_status.target.x                = [  0.960  0.218 -0.098  0.992 ]
sys_status.via.q                   = [ -0.829 -1.640  0.230  0.000 ]
sys_status.via.x                   = [  0.210  0.036 -1.580  1.501 ]
sys_status.via_step                = 1
sys_status.via_time                = 602049408.2
sys_status.col_status.has_collisions = 0
sys_status.col_status.link1        = -1
sys_status.col_status.link2        = -1
sys_status.col_status.obj1         = -1
sys_status.col_status.obj2         = -1
sys_status.rotation_cmd            = [ 1 -1 1 1 ]
sys_status.delta_via.dq            = [  0.000  0.000  0.000  0.000 ]
sys_status.delta_via.dx            = [  0.000  0.000  0.000  0.000 ]
sys_status.delta_via.mag           = 0.0000
sys_status.algo_code               = 2  RA_ALGO_CONTACT
sys_status.algo_code_override      = 0  RA_ALGO_NO_OVERRIDE
sys_status.delta_via_code          = 3  RA_DELTA_VIA_USER
sys_status.plane_angle             = 0.0097
sys_status.tau                     = [    0.0    0.0    0.0    0.0 ]
sys_status.delta_tau               = 0.0000
sys_status.track_state             = 0  RA_NO_TRACKING
sys_status.energy_limit            = 0.0
sys_status.energy_value            = 0.0
```

```
sys_status.time_limit              = 602049206.7
sys_status.grapple_state           = 0  Fingers-Open
sys_status.grapple_temp            = 99.5
sys_status.grapple_exp_time        = 0.000000
sys_status.grapple_phase           = 2  RA_OK_TO_OPEN
sys_status.mtr_volt_sign           = [ 0 0 0 0 ]
sys_status.actively_holding_position = 0
sys_status.time                    = 602050014.0
```

- sys_status.curr_limit – Joint current limits (mA)
- sys_status.mtr_volt – Joint motor voltages (V)
- sys_status.brake – Boolean for whether the brakes are applied on each joint when the command is complete
- sys_status.use_pots – Boolean for whether to use the pots on each joint for motion control
- sys_status.htr_on – Boolean for whether the heater is on for each joint
- sys_status.tool – Selected tool (blade 1 | blade 2 | IDC | scoop | wrist | grapple | grapple-attach)
- sys_status.cool_down – Boolean for whether cool down activity is active
- sys_status.fault_chk_mask – Fault mask code that is enabled for joint and misc. The value will be 0 if all faults are disabled
- sys_status.fault_status – Fault mask code for which faults have tripped for joint and misc. The value will be 0 if there are no faults
- sys_status.act_cmd – Last actuator command
- sys_status.act_args – Arguments for last actuator command
- sys_status.task_cmd – Last non-actuator command
- sys_status.task_args – Arguments for last non-actuator command
- sys_status.target.q – Target joint position in radians of the last RA_MOVE_GUARDED
- sys_status.target.x – Target Cartesian position and orientation of the last RA_MOVE_GUARDED
- sys_status.via.q – Joint position in radians of the current via point
- sys_status.via.x – Cartesian position and orientation of IDA
- sys_status.via_step – The via point that the IDA is moving towards
- sys_status.via_time – During a timed joint move, the time (sclkd) that the current via point started
- sys_status.col_status.has_collisions – Boolean for existence of collision
- sys_status.col_status.link1 – linkID of the RA link having collisions or -1 if none
- sys_status.col_status.link2 – linkID of the other link
- sys_status.col_status.obj1 – object ID of one object
- sys_status.col_status.obj2 – object ID of the other object
- sys_status.rotation_cmd – Two possible values to represent direction, -1 for negative direction and 1 for positive direction. If the IDA is moving, the value is current direction. If the IDA is not moving, the value is the last recorded direction.
- sys_status.delta_via.dq – Modification of joint position in radians based on dx
- sys_status.delta_via.dx – Modification of Cartesian position and orientation based on torque monitoring during accomodation
- sys_status.delta_via.mag – Magnitude of delta via
- sys_status.algo_code - Torque threshold (1=accomodate, 2=contact, 3=free space)
- sys_status.algo_code_override – Override or not override the torque threshold. If not set to override, then use algo code; otherwise, use this algo_code_override value
- sys_status.delta_via_code – Index (0=acq,1=dig,2=scrape,3=user) into parameter to determine direction of accommodation to limit elevated joint torques

- sys_status.plane_angle – When using a scrape command, angle (radians) of scrape relative to the x-y plane
- sys_status.tau – Joint torques (N-m)
- sys_status.delta_tau – Updated torque (N-m) from delta via computation, normalized value of the joint torque above its threshold
- sys_status.track_state – State of resource tracking: {RA_NO_TRACKING=0, RA_TRACK_ENERGY=1, RA_TRACK_TIME=2, RA_TRACK_BOTH=3}
- sys_status.energy_limit – If sys_status.track_statue is RA_TRACK_ENERGY, value is amount of energy (Watt-hours) not to exceed
- sys_status.energy_value – If sys_status.track_statue is RA_TRACK_ENERGY, value is amount of energy (Watt-hours) used
- sys_status.time_limit – If sys_status.track_statue is RA_TRACK_TIME, value is limit in minutes for the current command
- sys_status.grapple_state – Grapple state (0=in-between, 1=closed, 2=open, 3=broken) based on the four combinations of the limit switch states
- sys_status.grapple_temp – Grapple temperature (C)
- sys_status.grapple_exp_time – Time (sclkd) that the grapple stops heating
- sys_status.grapple_phase – Grapple phase (launch_locked | stowed | ok_to_open | seis_grappled | wts_grappled | hp3_grappled)
- sys_status.mtr_volt_sign – Motor voltage sign (1 is positive and -1 is negative) during active hold, 0 otherwise
- sys_status.actively_holding_position – Boolean for whether active hold is active
- sys_status.time – Time (sclkd) the system status values were recorded

5.1.2 IDA History

The history file contains the chronological concatenation of the last 300 status outputs (both arm and system) from the point when the dump history command is initialized. Details regarding IDA status can be found in Section 5.1.1.

In the history file, the point value goes in descending order starting at 299 and ends at 0. Also provided is the index value of the heap that stores the state and history data. Note that the heap size is 300 and the values will be overwritten once the heap is full. The point and index value are shown in the IDA history file as:

0 points before latest (index = 295):

5.1.3 IDA Parameters

The parameter file contains a timestamp (SCLK) and the following parameters:

5.1.3.1 RA_ARM_PARAMS

```
Parameter Set: RA_ARM_PARAMS
dh.a          = [  0.029609000  0.976664007  0.857989013  0.000000000 ]
dh.d          = [ -0.056456000 -0.140837997  0.000000000  0.000000000 ]
dh.alpha      = [  1.570799947  0.000000000  0.000000000  1.570799947 ]
x_blade_1     = [ -0.015010000  0.000000000  0.105240002  0.000000000 ]
x_blade_2     = [ -0.015010000  0.000000000  0.105240002  0.000000000 ]
x_rac         = [ -0.560000002 -0.055000000 -0.150000006  1.570799947 ]
```

```
x_scoop         = [  0.077880003  0.000000000  0.112910002  1.431169987 ]
x_wrist         = [  0.000000000  0.000000000  0.000000000  0.000000000 ]
x_grapple_attach = [  0.000000000 -0.040940002  0.000000000  3.141592979 ]
dist_grapple    = [  0.000000000  0.000000000  0.298999995  0.000000000 ]
dist_seis       = [  0.000000000  0.000000000  0.625000000  0.000000000 ]
dist_wts        = [  0.000000000  0.000000000  0.816999972  0.000000000 ]
dist_hp3        = [  0.000000000  0.000000000  0.732999980  0.000000000 ]
```

- dh.a – joint1, joint2, joint3, joint4 Denavit and Hartenberg (DH)[1] link lengths (m).
- dh.d – joint1, joint2, joint3, joint4 Denavit and Hartenberg (DH))[1] link offsets (m).
- dh.alpha – joint1, joint2, joint3, joint4 Denavit and Hartenberg (DH))[1] frame twists (rad).
- x_blade_1 – x, y, z, theta coordinates of the forward-facing scraping blade in the wrist frame (x, y, z in m and theta in rad).
- x_blade_2 – x, y, z, theta coordinates of the backward-facing scraping blade in the wrist frame (x, y, z in m and theta in rad).
- x_rac – x, y, z, theta coordinates of the IDC in the wrist frame (x, y, z in m and theta in rad).
- x_scoop – x, y, z, theta coordinates of the scoop blade in the wrist frame (x, y, z in m and theta in rad).
- x_wrist – x, y, z, theta coordinates of the wrist in the wrist frame (x, y, z in m and theta in rad).
- x_grapple_attach – x, y, z, theta coordinates of the grapple-attach frame in the wrist frame (x, y, z in m and theta in rad).
- dist_grapple – Element 2 (third element) is the distance between the Grapple-Attach frame and the Grapple frame (m). The other elements must be zero.
- dist_seis – Element 2 (third element) is the distance between the Grapple-Attach frame and the SEIS frame (m). The other elements must be zero.
- dist_wts – Element 2 (third element) is the distance between the Grapple-Attach frame and the WTS frame (m). The other elements must be zero.
- dist_hp3 - Element 2 (third element) is the distance between the Grapple-Attach frame and the HP3 frame (m). The other elements must be zero.

5.1.3.2 RA_JOINT_PARAMS

```
Parameter Set: RA_JOINT_PARAMS
q_min           = [ -3.5314     -2.3953     -0.4324     -0.1742     ]
q_max           = [  1.4585      1.307       4.0478      3.5887     ]
pot.b           = [  1394.31     2175.06     955.99      1491.77    ]
pot.m           = [ -683.776     685.114     691.49      681.93     ]
current.b       = [  2048        2048        2048        2048       ]
current.m       = [  0.00032143  0.00032143  0.00032143  0.00032143 ]
vref.b          = [  2048        2048                               ]
vref.m          = [  0.00225     0.00225                            ]
gear_ratio      = [  -12500      16000       10000       4000       ]
rad_per_enc     = [  3.1416      3.1416      3.1416      3.1416     ]
enc_offset      = [ -0.544255018  0.824948013  0.237636998  0.305449992 ]
htr_power       = [  14.1        3.1         11.2        11.2       ]
start_count     = [  5           3           3           3          ]
nonstart_count. = [  1           1           1           1          ]
direction_crnt  = [  85          85          85          85         ]
```

[1] See Appendix B for definition of Denavit and Hartenberg link frames

```
direction_delay  = [  50          50          50          50          ]
```

- q_min – joint1, joint2, joint3, joint4 minimum joint angles (rad).
- q_max – joint1, joint2, joint3, joint4 maximum joint angles (rad).
- pot.b – joint1, joint2, joint3, joint4 pot offset (LSBs). Digital number values that the PEB reports when the engineering unit value is 0 for the joint angle measured by the potentiometer.
- pot.m – joint1, joint2, joint3, joint4 pot slope (LSB/rad).
- current.b – joint1, joint2, joint3, joint4 motor current offset (LSBs). Current reference value when the joint angle is zero.
- current.m – joint1, joint2, joint3, joint4 motor current slope (amps/LSB).
- vref.b – vref1, vref2 voltage reference offset (LSBs). Voltage reference value when the joint angle is zero.
- vref.m – vref1, vref2 heater2 voltage reference slope (volts/LSB).
- gear_ratio – joint1, joint2, joint3, joint4 gear ratios
- rad_per_enc – joint1, joint2, joint3, joint4 radians per encoder pulse at the motor shaft. To convert to joint angle, divide by the *gear_ratio*.
- enc_offset – joint1, joint2, joint3, joint4 encoder offset (LSBs). This value was chosen such that when the encoder count is 1000 (this prevents rollover), the computed joint angle equals *q_min*. Also, even though the encoder is physically located on the motor shaft, *enc_offset* is expressed as if it were on the joint output. I.e., multiply by the *gear_ratio* to convert to the equivalent value at the motor shaft.
- htr_power – joint1, 2, and 3, joint4, grapple heater1, grapple heater2 heater power (watts)
- start_count – joint1, joint2, joint3, joint4 the number of current samples the firmware ignores immediately after the motor is powered on. During this time zero current will be reported in the response. A value of FF will disable overcurrent monitoring and will cause zero current to be reported in the response, *always*.
- nonstart_count – joint1, joint2, joint3, joint4 the number of consecutive overcurrent samples the firmware ignores at any time after the startup count. A value of FF will disable overcurrent monitoring.
- direction_crnt – joint1, joint2, joint3, joint4 current (mA) above which the encoder count direction is reversed when the motor is stopped. Below this value the encoder counter will continue to count in the same direction when the motor is stopped.
- direction_delay – joint1, joint2, joint3, joint4 time delay (ms) after which the encoder count direction is reversed when the motor is stopped.

5.1.3.3 RA_CTRL_PARAMS

```
Parameter Set: RA_CTRL_PARAMS
mtr_volt_max          = [  30       30       30       30       ]
mtr_volt_min          = [  1        1        1        1        ]
mtr_volt_init         = [  0        0        0        0        ]
acc_max               = [  0.013962  0.010908  0.017453  0.043633  ]
vel_max               = [  0.020944  0.016362  0.02618   0.06545   ]
vel_min               = [  0        0        0        0        ]
eps_joint             = [  0.03     0.03     0.03     0.05     ]
deadband_strt         = [  3        3        3        3        ]
deadband_final_via_only  =  1
deadband_end          = [  0.001    0.001    0.001    0.001    ]
eps_time              =  0.4
p_gain                = [  2750     3520     2200     880      ]
```

```
i_gain                 = [  2291.7    2933.3    1833.3    733.33    ]
d_gain                  = [  0        0         0         0         ]
filter_p_gain           =  0.0025
filter_i_gain          =  0.0127
loop_time               =  0.1
loop_time_thermostat    =  10
loop_time_cooldown      =  0.2
delta_via_period        =  4
paci_timeout            =  1000
stop_to_no_op_time_delay =  5
power_on_time_delay     =  1
off_count               =  2
```

- mtr_volt_max – joint1, joint2, joint3, joint4 max allowable motor voltage (volts).
- mtr_volt_min – joint1, joint2, joint3, joint4 min allowable motor voltage (volts).
- mtr_volt_init – joint1, joint2, joint3, joint4 volts applied to motor to start move (volts).
- acc_max – joint1, joint2, joint3, joint4 trapezoidal profile default accel(rad/s^2)
- vel_max – joint1, joint2, joint3, joint4 maximum joint velocity (rad/s)
- vel_min – joint1, joint2, joint3, joint4 minimum joint velocity (rad/s)
- eps_joint - joint1, joint2, joint3, joint4 when all joints get to within *eps_joint* radians of their commanded via point, the next via point is sent to the RAE.
- deadband_strt – joint1, joint2, joint3, joint4 joint deadband at cmd start. Joint moves less than deadband_strt will be ignored. Units are LSB of encoder counts or raw potentiometer reading depending on the joint sensor being used for ctrl (default is encoder).
- deadband_final_via_only – Bool for checking whether jnt is within deadband_end of via point at final via point only (1) or for each via point (0)
- deadband_end – joint1, joint2, joint3, joint4 at cmd start. Final position that will be less than deadband_end will be considered successful.
- eps_time – when all joints get to within *eps_time* seconds of their commanded via point, the next via point is sent to the RAE. Applicable only to timed moves.
- p_gain - joint1, joint2, joint3, joint4 joint controller proportional gain (V/rad).
- i_gain - joint1, joint2, joint3, joint4 joint controller integral gain (V/sec).
- d_gain – joint1, joint2, joint3, joint4 joint controller derivative gain (Vsec/rad).
- filter_p_gain – accommodation algorithm PI filter proportional gain.
- filter_i_gain – accommodation algorithm PI filter integral gain.
- loop_time – controller sampling period (sec) for all commands except for the IDA_THERMOSTAT command.
- loop_time_thermostat – controller sampling period (sec) for the IDA_THERMOSTAT command.
- loop_time_cooldown – controller sampling period (sec) during actuator cooldown.
- delta_via_period – period to send delta via (the change to the commanded via point) to RAE when the accommodation algorithm is executing.
- paci_timeout – timeout for paci communication to RAE (milliseconds).
- stop_to_no_op_time_delay – time delay for collecting final data after a stop command (sec).
- power_on_time_delay – time delay between eps power-on request and communicating with the RAE (sec).
- off_count - number of consecutive samples with motor power off before declaring move complete.

5.1.3.4 RA_LANDER_TO_RA

```
Parameter Set: RA_LANDER_TO_RA
frame.x     = [ 0 0 0   ]
frame.q     = [ 1 0 0 0 ]
```

- frame.x – x, y, z of the RA base frame in the payload coordinate system (m).
- frame.q - the unit quaternion representing the orientation transformation from the payload frame to the RA base frame. The identity is represented by [1 0 0 0].

5.1.3.5 RA_PATH_PARAMS

```
Parameter Set: RA_PATH_PARAMS
path.cart.type  =  0
path.cart.error  =  0.01
path.cart.delta  =  0.05
path.joint_delta =  0.085
path.time_delta  =  5
```

- path.cart.type – Type of motion.
    - 0 – straight-line motion.
    - 1 – joint-interpolation motion.
- path.cart.error – allowable path error (m) for straight-line Cartesian motion.
- path.cart.delta – allowable delta (m) between via points .
- path.joint_delta – allowable delta (rad) between via points for spatial joint moves.
- path.time_delta – allowable delta (sec) between via points for timed joint moves.

5.1.3.6 RA_TASK_ALGO

```
Parameter Set: RA_TASK_ALGO
[op_code  algo_code  delta_via_code] = [ 11  3   3 ]
[op_code  algo_code  delta_via_code] = [ 13  1   1 ]
[op_code  algo_code  delta_via_code] = [ 18  3   3 ]
[op_code  algo_code  delta_via_code] = [ 22  1   2 ]
[op_code  algo_code  delta_via_code] = [ 47  3   3 ]
[op_code  algo_code  delta_via_code] = [ 48  3   3 ]
[op_code  algo_code  delta_via_code] = [ 49  2   3 ]
[op_code  algo_code  delta_via_code] = [ 0   0   0 ]
[op_code  algo_code  delta_via_code] = [ 0   0   0 ]
[op_code  algo_code  delta_via_code] = [ 0   0   0 ]
[op_code  algo_code  delta_via_code] = [ 0   0   0 ]
[op_code  algo_code  delta_via_code] = [ 0   0   0 ]
```

- op_code – the opcode for the command to which the algorithm applies. See rat_show_codes.
- algo_code – the control algorithm to be used during execution of the command with *op_code*.
    - 1 - accommodate - the arm will modify the trajectory in the direction of *delta_via* (e.g., follow the surface with which it is in contact) when any joint torque exceeds *thresh_a*.
    - 2 - contact - the arm will stop on contact when any joint torque exceeds *thresh_c*.
    - 3 - free space - the arm will stop on contact when any joint torque exceeds *thresh_f*.
- delta_via_code – which *delta_via* to use when the *accommodate* algorithm is active

- 1 - dig - delta_via is in direction orthogonal to the direction of motion, upward, and in the plane of the motion as determined by *ida_trench_angle*.
- 2 - scrape - delta_via is orthogonal to, and upward from, the scraping plane found during IDA_SCRAPE.
- 3 - user - delta_via_user.

5.1.3.7 RA_ACT_ALGO

```
Parameter Set: RA_ACT_ALGO
[op_code  algo_code  delta_via_code] =  [  17  2   3 ]
[op_code  algo_code  delta_via_code] =  [  19  2   3 ]
[op_code  algo_code  delta_via_code] =  [  20  3   3 ]
delta_via_user                        =  [  0   0  -1 ]
dvia_thresh                          =  0.01
```

- op_code – the opcode for the command to which the algorithm applies. See rat_show_codes.
- algo_code – the control algorithm to be used during execution of the command with op code *op_code*.
  - 1 - accommodate - the arm will modify the trajectory in the direction of *delta_via* (e.g., follow the surface with which it is in contact) when any joint torque exceeds *thresh_a*.
  - 2 - contact - the arm will stop on contact when any joint torque exceeds *thresh_c*.
  - 3 - free space - the arm will stop on contact when any joint torque exceeds *thresh_f*.
- delta_via_code – which *delta_via* to use when the *accommodate* algorithm is active
  - 1 - dig - delta_via is in direction orthogonal to the direction of motion, upward, and in the plane of the motion as determined by *ida_trench_angle*.
  - 2 - scrape - delta_via is orthogonal to, and upward from, the scraping plane found during IDA_SCRAPE.
  - 3 - user - delta_via_user.
- delta_via_user – x, y, z
- dvia_thresh – threshold (magnitude of Cartesian *delta_via* in m) at which accommodation begins.

5.1.3.8 RA_TAU_PARAMS

```
Parameter Set: RA_TAU_PARAMS
limit        =  [  26        91        53        10              ]
thresh_a     =  [  26        91        53        10              ]
thresh_c     =  [  35        120       65        10.5            ]
thresh_f     =  [  35        120       65        10.5            ]
thresh_d     =  [  29.125    37.28     23.3      9.32            ]
fit_nl       =  [  2         2         2         2               ]
c_nl_0       =  [  0.014134   0.016033   0.014041   0.0054075    ]
c_nl_1       =  [  0.0021647  0.0008571  0.0007949  0.0078765    ]
c_nl_2       =  [  0.028697   0.052189   0.052205   0.0078644    ]
c_nl_3       =  [  0.0081567  0.0115545  0.009798   0.0041967    ]
c_nl_4       =  [  0.01211    0.024053   0.020149   0            ]
temp.b       =  [  2525      2525      2525      2525      613.327 ]
temp.m       =  [  0.2266    0.2259    0.2266    0.2283    0.4469  ]
mtr_volt_max =  [  30        30        30        30              ]
```

- limit – joint1, joint2, joint3, joint4 joint torque limits (N-m) above which damage may occur.

- thresh_a – joint1, joint2, joint3, joint4 joint torque thresholds (N-m) for the accommodation algorithm.
- thresh_c – joint1, joint2, joint3, joint4 joint torque thresholds (N-m) for the contact algorithm.
- thresh_f – joint1, joint2, joint3, joint4 joint torque thresholds (N-m) for the free-space algorithm.
- thresh_d – joint1, joint2, joint3, joint4 detent torque in N-m added to threshold if actuator moving less than detent velocity.
- fit_nl – joint1, joint2, joint3, joint4 no-load current curve fit type
  - 0 - Polynomial fit
  - 1 - Exponential fit
  - 2 - Exponential fit w/ motor volt
- c_nl_0 – joint1, joint2, joint3, joint4 No-load current curve fit first coefficient
- c_nl_1 – joint1, joint2, joint3, joint4 No-load current curve fit second coefficient
- c_nl_2 – joint1, joint2, joint3, joint4 No-load current curve fit third coefficient
- c_nl_3 – joint1, joint2, joint3, joint4 No-load current curve fit forth coefficient
- c_nl_4 – joint1, joint2, joint3, joint4 No-load current curve fit fifth coefficient
- temp.b – joint1, joint2, joint3, joint4 grapple temperature offset (LSBs). Value at which temperature is zero degrees Celsius.
- temp.m – joint1, joint2, joint3, joint4, grapple temperature slope (Celsius/LSB).
- mtr_volt_max - joint1, joint2, joint3, joint4 max motor volt in volts used during joint actuator characterization.

5.1.3.9 RA_TAU_PARAMS2

```
Parameter Set: RA_TAU_PARAMS2
current_limit = [ 110    260    210    110    ]
torque_limit  = [ 35     120    65     10.5   ]
no_load_volt  = [ 0      0      0      0      ]
current_delta = [ 1      1      1      1      ]
tau_detent    = [ 29.125 37.28  23.3   9.32   ]
speed_detent  = [ 0.018  0.0098 0.0158 0.048  ]
fit_ka        = [ 1      1      1      1      ]
c_ka_0        = [ 468.72 574.79 371.6  116.53 ]
c_ka_1        = [ 0      0      0      0      ]
c_ka_2        = [ 0      0      0      0      ]
c_ka_3        = [ 0      0      0      0      ]
c_ka_4        = [ 0      0      0      0      ]
```

- current_limit – joint1, joint2, joint3, joint4 default current limits (mA).
  - If current_limit = -1 - compute current limit autonomously.
- torque_limit – joint1, joint2, joint3, joint4 torque limit (N-m) for computing current limit as function of temperature.
- no_load_volt – joint1, joint2, joint3, joint4 voltage (V) used with voltage-based no-load model (if selected) for computing current limit as function of temperature.
- current_delta – joint1, joint2, joint3, joint4 computed torque is zero if current is within +/- current_delta (LSB) of current offset b.
- tau_detent – joint1, joint2, joint3, joint4 detent torque (N-m) for computing detent current.
- speed_detent – joint1, joint2, joint3, joint4 speed below which detent comes into play (rad/s).
- fit_ka – joint1, joint2, joint3, joint4 torque constant curve fit type.
  - 0 – Polynomial Fit
  - 1 – Exponential Fit

- c_ka_0 - joint1, joint2, joint3, joint4 torque constant curve fit first coefficient
- c_ka_1 - joint1, joint2, joint3, joint4 torque constant curve fit second coefficient
- c_ka_2 - joint1, joint2, joint3, joint4 torque constant curve fit third coefficient
- c_ka_3 - joint1, joint2, joint3, joint4 torque constant curve fit forth coefficient
- c_ka_4 - joint1, joint2, joint3, joint4 torque constant curve fit fifth coefficient

5.1.3.10 RA_SCIENCE_DATA_PERIOD

```
Parameter Set: RA_SCIENCE_DATA_PERIOD
science data period =  1
```

- science_data_period - integer number of sampling periods between collection of each set of science data which are stored in heap.

5.1.3.11 RA_FAULT_DETECT

```
Parameter Set: RA_FAULT_DETECT
enc_pot_delta      = [  0.1    0.1    0.1    0.1     ]
runaway_joint_delta =  0.1745
runaway_pos_delta   =  0.1
runaway_ori_delta   =  3.14
runaway_time_delta  =  5
joint_delta        = [  0.034  0.034  0.034  0.034    ]
vref_nominal       = [  4.5   -4.5                ]
vref_delta         = [  0.03   0.03               ]
enc_delta          =  2
pot_delta          =  20
tau                = [  10    29.2   17.2   3       ]
motor_samples      =  100
current_epsilon    =  1
heater_time        = [  1200   1200   180    180     ]
temp_delta         = [  20     20     20     20    5 ]
impeded_samples    = [  2      2      2      2       ]
pot_samples        =  20
encoder_samples    =  20
enc_pot_samples    =  2
vref_samples       =  3
runaway_samples    =  2
kinematic_samples  =  2
heater_rtd_samples =  3
htr_crnt           = [  120    80     200    200     ]
htr_crnt_offset    =  2048
ace_cal_samples    =  3
power_30v_samples  =  10
stopped_samples    =  100
stalled_samples    =  100
deflection_limit   =  0.5
subtrench_attempts =  2
dvia_mag_limit     =  0.4
default_cmd_token  =  1
collision_mode     =  2
```

lander_tol = 0.015

- enc_pot_delta – joint1, joint2, joint3, joint4 allowable difference in radians between the joint angle computed from the encoder and the pot above which a fault is declared.
- runaway_joint_delta – joint delta for a spatial joint move, if the difference in radians between the commanded and current joint angle is greater than *runaway_joint_delta*+ *joint_delta*+*eps_joint,* then a fault is declared.
- runaway_pos_delta – position delta for a Cartesian move, if the difference in meters between the commanded and current Cartesian position is greater than *runaway_pos_delta*+ cart.delta, then a fault is declared.
- runaway_ori_delta – orientation delta for a Cartesian move, if the difference in radians between the commanded and current orientation is greater than *runaway_ori_delta*, then a fault is declared.
- runaway_time_delta – time delta for a timed joint move, if the difference in seconds between the current and commanded time is greater than *runaway_time_delta*+ time_delta, then a fault is declared.
- joint_delta – joint1, joint2, joint3, joint4 joint delta where if a joint is commanded to move or moves within *joint_delta* radians of its joint hard stop, then a kinematic fault is declared.
- vref_nominal – vref1, vref2 nominal values for the positive and negative voltage references.
- vref_delta – vref1, vref2 voltage delta where if a voltage reference differs from *vref_nominal* by more than *vref_delta*, then a fault is declared.
- enc_delta – encoder delta where if a motor is on and the encoder counter changes by less than *enc_delta* counts between samples, then an encoder fault is declared.
- pot_delta – potentiometer delta where if a motor is on and the joint torque is less than *tau* and the pot changes by less than *pot_delta* LSBs between samples, then an pot fault is declared.
- tau – joint1, joint2, joint3, joint4 torque threshold for pot check (N-m).
- motor_samples – max allowable consecutive anomalies after which a motor fault is declared.
- current_epsilon – current in LSBs below *current_epsilon* is considered to be zero. Used in motor, motor stall, and 30V power converter fault detection algorithms.
- heater_time – heater time for when a heater is turned on, the heating rate is checked after *heater_time* seconds elapses.
    - Element 0 is for joint 1, 2, and 3 heaters
    - Element 1 is for joint 4 heater
    - Element 2 is for grapple heater 1
    - Element 3 is for grapple heater 2
- temp_delta – temperature delta for when a heater is turned on, if the joint temperature (i = 0, 3) has not risen by *temp_delta* LSBs in *heater_time* seconds, then a heater or temperature sensor fault is declared.
    - Element 0 is for joint 1
    - Element 1 is for joint 2
    - Element 2 is for joint 3
    - Element 3 is for joint 4
    - Element 4 is for the grapple
- impeded_samples – joint1, joint2, joint3, joint4 max allowable consecutive anomalies after which a motion-impeded event is declared.
- pot_samples - max allowable consecutive anomalies after which a pot fault is declared.
- encoder_samples – max allowable consecutive anomalies after which an encoder fault is declared.
- enc_pot_samples – max allowable consecutive anomalies after which an encoder-pot delta fault is declared.

- vref_samples – max allowable consecutive anomalies after which a voltage reference fault is declared.
- runaway_samples – max allowable consecutive anomalies after which a joint runaway fault is declared.
- kinematic_samples – max allowable consecutive anomalies after which a kinematic fault is declared.
- heater_rtd_samples – max allowable consecutive anomalies after which a heater or temperature sensor fault is declared.
- htr_crnt – if a heater is turned on and the heater current in LSBs is less than *htr_crnt*, then a heater/sensor fault is declared. Also used in the 30V power supply fault detection algorithm.
    - Element 0 is for joint 1, 2, and 3 heaters
    - Element 1 is for joint 4 heater
    - Element 2 is for grapple heater 1
    - Element 3 is for grapple heater 2
- htr_crnt_offset – zero heater current in LSBs.
- ace_cal_samples – max allowable consecutive anomalies after which an A/D converter calibration fault is declared.
- power_30V_samples – max allowable consecutive anomalies after which a 30V power converter fault is declared.
- stopped_samples – max allowable consecutive anomalies after which a premature motion termination fault is declared.
- stalled_samples – max allowable consecutive anomalies after which a motor stalled fault is declared.
- deflection_limit – max allowable joint delta between deflected and undeflected pose (RSS) in radians.
- subtrench_attempts – max allowable attempts at digging the same subtrench before issuing fault.
- dvia_mag_limit – max allowable delta via (m).
- default_cmd_token – default command ID token for RA_SEND_DATA. Used primarily for sending of ra_history data when a fault occurs.
- collision_mode – mode used for calculation of link poses in collision detection:
    - 0 – collision detection off.
    - 1 – No deflection compensation in collision detection.
    - 2 – Use deflection compensation in collision detection.
- lander_tol - the tolerance to add around each lander object whose individual tolerance is not specified for detecting collisions in m.

5.1.3.12 RA_TASK_C

```
Parameter Set: RA_TASK_C
calibrate.home    = [ -3.313 -2.21 -0.401  0.2 ]
calibrate.angle   = [  0.2    0.2   0.2    0.2 ]
calibrate.current = [  30     120   40     18  ]
calibrate.time    = [  40     40    40     40  ]
calibrate.delta   = [  0      0     0      0   ]
```

- calibrate.home – joint1, joint2, joint3, joint4 staging angles if call to IDA_CALIBRATE is set with the enum RA_NEG_HARDSTOP as argument for each joint.
- calibrate.angle – joint1, joint2, joint3, joint4 angles from the hard stop the joints will move to prior to the timed moves that calibrate the arm.

- calibrate.current – **Obsolete. The current limits are set autonomously at the actuator command level.**
- calibrate.time – joint1, joint2, joint3, joint4 times in seconds of timed moves which run joints up against hard stops.
- calibrate.delta – joint1, joint2, joint3, joint4 difference in radians between pot and encoder at hard stops. It is added to the hard stop joint angle computed from the pots prior to setting the encoder counters. *delta* compensates for the windup in the drive train which occurs when the joints run up against the hard stops.

5.1.3.13 RA_TASK_DT

```
Parameter Set: RA_TASK_DT
dig_trench.height              = -0.2
dig_trench.[depth width length] =  [  0.002  0.073  0.25 ]
dig_trench.corner              =  0.15
dig_trench.scoop_angle          =  0.125
dig_trench.carry_angle         =  2.5
dig_trench.dump_angle           =  7.7125
dig_trench.return_angle        =  2
dig_trench.way_point           = [  0.1    0.1    0    ]
```

- dig_trench.height - height in meters above the starting commanded position to where the scoop is initially moved. It is added to ra_pos_z.
- dig_trench.depth - the depth in meters of a single subtrench. If this parameter is less than or equal to the trench_depth specified in the command, then more than one subtrench in depth will be dug.
- dig_trench.width - the width in meters of a single subtrench. If this parameter is less than or equal to the trench_width specified in the command, then more than one subtrench in width will be dug. Note that the width of the scoop is 0.073m.
- dig_trench.length - the length in meters of a single subtrench. The number of subtrenches dug in length at each depth is a function of trench_length, trench_depth, dig_angle1, dig_angle2, and depth.
- dig_trench.corner - length in meters of the corner motion. The corner motion is the translation and rotation that the scoop makes from the entry ramp to the trench bottom and the from the trench bottom to the exit ramp. If the length of the subtrench is less than the corner motion, then the corner motion from the entry ramp to the trench bottom is proportionally reduced by corner/subtrench_length in both translation and rotation. The full corner motion is always executed from the bottom of the trench to the exit ramp. The parameter should be large enough so that the corner motion can complete without the back of the scoop hitting the entry ramp during the move from the entry ramp to the trench bottom or hitting the trench bottom during the move from trench bottom to the exit ramp. The default value is 0.15m.
- dig_trench.scoop_angle - planning angle for the scoop in radians. It is the angle from the direction of motion of the scoop to the scoop frame approach (z) vector during digging motions (entry ramp, trench bottom, exit ramp).
- dig_trench.carry_angle - scoop angle in radians during the move from the exit of the subtrench to the dump position.
- dig_trench.dump_angle - scoop angle in radians after dumping the contents.
- dig_trench.return_angle - scoop angle in radians upon returning to way point post-dump.
- dig_trench.way_point – x, y, z intermediate position expressed relative to the dump position in meters through which the scoop passes enroute from the exit of the subtrench to the dump position.

5.1.3.14 RA_TASK_SC

```
Parameter Set: RA_TASK_SC
scrape.guard_angle   =  1.629
scrape.guard_depth   =  0.1
scrape.rake          = [ 0  0  1.5368]
scrape.backoff       =  0.06
scrape.height_initial =  0
```

- scrape.guard_angle - scoop angle in radians for the guarded move.
- scrape.guard_depth - attempted depth in meters of the guarded move.
- scrape.rake – blade1, blade2, scoop rake angle in radians.
- scrape.backoff - distance in meters to back away from the surface prior to each scraping motion to allow set up to the other tool.
- scrape.height_initial - initial height in meters above the surface where the tool is placed prior to initiation of the scraping motion.

5.1.3.15 RA_COMPRESS_PARAMS

```
Parameter Set: RA_COMPRESS_PARAMS
compress.wSyncInterval =  1024
compress.wBlockSize   =  16
compress.wSegmentSize  =  1024
compress.type         =  0
```

- scrape.wSyncInterval – sync interval length used in Golomb-Rice encoding algorithm.
- scrape.wBlockSize - block size used in Golomb-Rice encoding algorithm.
- scrape.wSegmentSize – segment size used in Golomb/Rice encoding algorithm.
- scrape.type – **Not used. All compression is Golomb-Rice**

5.1.3.16 RA_MAX_FAULTS

```
Parameter Set: RA_MAX_FAULTS
max_fault.ace_reset     =  0
max_fault.motion_impeded =  4
max_fault.file_io       =  0
max_fault.paci          =  0
max_fault.tv_power      =  0
max_fault.stopped       =  3
max_fault.stalled       =  3
max_fault.overheat      =  3
```

- max_fault.ace_reset – number of allowed micro-processor reset faults before safing the RA.
- max_fault.motion_impeded - number of allowed arm movement obstructed faults before safing the RA.
- max_fault.file_io - number of allowed reading or writing to a file faults before safing the RA.
- max_fault.paci - number of allowed no communications the RAE faults before safing the RA.
- max_fault.tv_power - number of allowed 30V power supply fault faults before safing the RA.

- max_fault.stopped - number of allowed premature motion termination faults before safing the RA.
- max_fault.stalled - number of allowed motor stalled faults before safing the RA.
- max_fault.overheat - number of allowed actuator over heat faults before safing the RA.

5.1.3.17 RA_FAULT_PARAMS

```
Parameter Set: RA_FAULT_PARAMS
fault_params.dt_backup              =  0.03
fault_params.scr_initial            =  0.005
fault_params.scr_delta              =  0.001
fault_params.scr_total              =  0.003
fault_params.ignore_col_blade_1       =  0
fault_params.ignore_col_blade_2       =  0
fault_params.ignore_col_rac          =  0
fault_params.ignore_col_scoop         =  0
fault_params.ignore_col_wrist        =  0
fault_params.ignore_col_grapple_attach =  0
fault_params.ignore_col_grapple       =  0
fault_params.ignore_col_seis         =  0
fault_params.ignore_col_wts          =  0
fault_params.ignore_col_hp3          =  0
```

- fault_params.dt_backup - distance in meters to move scoop away from trench when recovering from a motion-impeded event during IDA_DIG_TRENCH.
- fault_params.scr_initial – distance to move to in meters from trouble spot as setup position for scraping. (not used)
- fault_params.scr_delta - change in meters in depth for each successive scrape. (not used)
- fault_params.scr_total - total change in meters in depth at which point scraping is terminated. (not used)
- fault_params.ignore_col_blade_1 - ignore collision mask forward-facing scraping blade.
- fault_params.ignore_col_blade_2 - ignore collision mask backward-facing scraping blade.
- fault_params.ignore_col_rac – boolean to ignore collision mask rac.
- fault_params.ignore_col_scoop - boolean to ignore collision mask scoop blade.
- fault_params.ignore_col_wrist - boolean to ignore collision mask wrist.
- fault_params.ignore_col_grapple_attach - boolean to ignore collision mask grapple-attach.
- fault_params.ignore_col_grapple - boolean to ignore collision mask grapple.
- fault_params.ignore_col_seis - boolean to ignore collision mask SEIS.
- fault_params.ignore_col_wts - boolean to ignore collision mask WTS.
- fault_params.ignore_col_hp3 - boolean to ignore collision mask HP3.

5.1.3.18 RA_POWER_STATES

```
Parameter Set: RA_POWER_STATES
power_states.power =  [  11  18  43]
eps_switch        =  0
```

- power_states.power – RA_POWER_IDLE state, RA_POWER_HEAT state, RA_POWER_MOVING state power value for each state in Watts.

- eps_switch - whether to use primary or secondary EPS switch. 0 for primary, 1 for secondary.

5.1.3.19 RA_GRAVITY

```
Parameter Set: RA_GRAVITY
gravity_params.acc_g    =  3.711
gravity_params.grav_vect =  [0.046825 -0.051123 0.997594]
```

- gravity_params.acc_g - gravitational acceleration magnitude (m/s^2).
- gravity_params.grav_vect – x, y, z gravitational vector in payload frame.

5.1.3.20 RA_DEFL_GENERAL

```
Parameter Set: RA_DEFL_GENERAL
defl_params.enable_abs     =  1
defl_params.enable_rel     =  0
defl_params.enable_tool    =  0
defl_params.enable_jnt_abs =  0
defl_params.enable_jnt_rel =  0
seis_mass                  =  8.900000
wts_mass                   =  7.300000
hp3_mass                   =  2.700000
compliance_scale           =  1.092301
```

- defl_params.enable_abs – boolean to enable deflection compensation algorithm for Cartesian moves.
- defl_params.enable_rel - boolean to enable deflection compensation algorithm for relative Cartesian moves. **(Not implemented)**
- defl_params.enable_tool - boolean to enable deflection compensation algorithm for tool-frame Cartesian moves. **(Not implemented)**
- defl_params.enable_jnt_abs - boolean to enable deflection compensation algorithm for absolute joint moves. **(Not implemented)**
- defl_params.enable_jnt_rel - boolean to enable deflection compensation algorithm for relative joint moves. **(Not implemented)**
- seis_mass – mass of the SEIS instrument in kilograms.
- wts_mass – mass of the WTS instrument in kilograms.
- hp3_mass – mass of the HP3 instrument in kilograms.
- compliance_scale – scaling factor for compliance parameters.

5.1.3.21 RA_DEFL_LINK_0 – Arm Base

```
Parameter Set: RA_DEFL_LINK_0
mass =  0
cm_x =  0
cm_y =  0
cm_z =  0
vCf  =  [  5.710000e-09 -5.620000e-25 -3.010000e-24
          -1.230000e-24  5.710000e-09 -2.060000e-24
           2.810000e-24  1.240000e-24  5.710000e-09 ]
wCf  =  [ -9.560000e-26 -1.980000e-23  7.580000e-25
```

```
        3.290000e-24  2.430000e-25  4.560000e-24
        9.160000e-24  6.570000e-24  4.230000e-25 ]
vCm  =  [  3.020000e-25  4.940000e-24  7.990000e-24
       -3.060000e-24  3.750000e-25  7.010000e-24
        4.930000e-25  5.290000e-24 -8.830000e-26 ]
wCm =   [  8.850000e-08 -2.280000e-25  7.170000e-24
        1.050000e-23  8.850000e-08 -5.630000e-24
        7.840000e-24 -6.390000e-24  8.850000e-08 ]
```

- mass – lumped mass for the link in kilograms.
- cm_x – x coordinate of the center of mass for the link.
- cm_y – y coordinate of the center of mass for the link.
- cm_z – z coordinate of the center of mass for the link.
- vCf – 9 values representing the 3x3 vCf component of the compliance matrix. These are in row-major order.
- wCf – 9 values representing the 3x3 wCf component of the compliance matrix. These are in row-major order.
- vCm – 9 values representing the 3x3 vCm component of the compliance matrix. These are in row-major order.
- wCm – 9 values representing the 3x3 wCm component of the compliance matrix. These are in row-major order.

5.1.3.22 RA_DEFL_LINK_1 – Azimuth

```
Parameter Set: RA_DEFL_LINK_1
mass =  0.509
cm_x =  0.025
cm_y =  0.0579
cm_z = -0.0556
vCf  =  [  3.840000e-08  2.310000e-08 -2.730000e-21
        2.310000e-08  2.250000e-08 -2.010000e-21
       -4.050000e-21 -2.900000e-21  3.060000e-07 ]
wCf  =  [  9.620000e-22 -5.420000e-09 -8.470000e-07
        5.420000e-09  1.060000e-19 -9.820000e-06
        8.470000e-07  6.030000e-07 -7.470000e-20 ]
vCm  =  [ -2.520000e-20  5.420000e-09  8.470000e-07
       -5.420000e-09  9.950000e-20  6.030000e-07
       -8.470000e-07 -9.820000e-06 -1.050000e-19 ]
wCm  =  [  2.220000e-05  5.130000e-18  1.820000e-20
        6.060000e-18  3.603190e-04  3.840000e-18
       -6.660000e-19  3.670000e-18  2.220000e-05 ]
```

- mass – lumped mass for the link in kilograms.
- cm_x – x coordinate of the center of mass for the link.
- cm_y – y coordinate of the center of mass for the link.
- cm_z – z coordinate of the center of mass for the link.
- vCf – 9 values representing the 3x3 vCf component of the compliance matrix. These are in row-major order.
- wCf – 9 values representing the 3x3 wCf component of the compliance matrix. These are in row-major order.

- vCm – 9 values representing the 3x3 vCm component of the compliance matrix. These are in row-major order.
- wCm – 9 values representing the 3x3 wCm component of the compliance matrix. These are in row-major order.

5.1.3.23 RA_DEFL_LINK_2 – Upper Arm

```
Parameter Set: RA_DEFL_LINK_2
mass =  1.2381
cm_x =  0.708
cm_y =  0
cm_z = -0.0762
vCf  =  [  6.180000e-06 -4.110000e-06  6.530000e-05
          -4.110000e-06  1.730570e-04  5.340000e-07
           6.530000e-05  5.340000e-07  7.940660e-04 ]
wCf  =  [ -3.250000e-07  8.270000e-05  7.000000e-06
          -7.690000e-05  3.250000e-07 -8.573120e-04
          -6.870000e-06  2.152100e-04 -6.990000e-12 ]
vCm  =  [ -3.250000e-07 -7.690000e-05 -6.870000e-06
           8.270000e-05  3.250000e-07  2.152100e-04
           7.000000e-06 -8.573120e-04 -6.990000e-12 ]
wCm  =  [  1.085949e-03  4.260000e-06 -4.010000e-09
           4.260000e-06  1.008957e-03 -2.210000e-10
         - 4.010000e-09 -2.210000e-10  3.512170e-04 ]
```

- mass – lumped mass for the link in kilograms.
- cm_x – x coordinate of the center of mass for the link.
- cm_y – y coordinate of the center of mass for the link.
- cm_z – z coordinate of the center of mass for the link.
- vCf – 9 values representing the 3x3 vCf component of the compliance matrix. These are in row-major order.
- wCf – 9 values representing the 3x3 wCf component of the compliance matrix. These are in row-major order.
- vCm – 9 values representing the 3x3 vCm component of the compliance matrix. These are in row-major order.
- wCm – 9 values representing the 3x3 wCm component of the compliance matrix. These are in row-major order.

5.1.3.24 RA_DEFL_LINK_3_STOWED – Forearm (w/ Grapple stowed)

```
Parameter Set: RA_DEFL_LINK_3_STOWED
mass =  1.9854
cm_x =  0.522
cm_y =  0.026
cm_z = -0.024
vCf  = [  1.020000e-07 -1.720000e-26 -4.830000e-08
          3.940000e-28  2.526130e-04 -2.830000e-24
         -4.830000e-08  9.850000e-27  1.019379e-03 ]
wCf  = [ -4.220000e-26 -5.870000e-08  8.920000e-22
          5.950000e-08 -1.240000e-26 -1.275284e-03
```

```
        5.940000e-28  3.824210e-04 -4.240000e-24 ]
vCm  = [ -4.290000e-31  5.950000e-08 -3.390000e-26
       -5.870000e-08  3.540000e-24  3.824210e-04
        6.670000e-29 -1.275284e-03  1.940000e-26 ]
wCm  = [  1.969914e-03 -1.120000e-21  9.440000e-09
       -9.370000e-29  1.791046e-03 -2.440000e-26
        9.440000e-09  5.300000e-24  7.513490e-04 ]
```

- mass – lumped mass for the link in kilograms.
- cm_x – x coordinate of the center of mass for the link.
- cm_y – y coordinate of the center of mass for the link.
- cm_z – z coordinate of the center of mass for the link.
- vCf – 9 values representing the 3x3 vCf component of the compliance matrix. These are in row-major order.
- wCf – 9 values representing the 3x3 wCf component of the compliance matrix. These are in row-major order.
- vCm – 9 values representing the 3x3 vCm component of the compliance matrix. These are in row-major order.
- wCm – 9 values representing the 3x3 wCm component of the compliance matrix. These are in row-major order.

5.1.3.25 RA_DEFL_LINK_3_UNSTOWED – Forearm (w/ Grapple Unstowed)

```
Parameter Set: RA_DEFL_LINK_3_UNSTOWED
mass =  1.4944
cm_x =  0.481
cm_y =  0.031
cm_z = -0.017
vCf  = [  1.020000e-07 -1.720000e-26 -4.830000e-08
        3.940000e-28  2.526130e-04 -2.830000e-24
       -4.830000e-08  9.850000e-27  1.019379e-03 ]
wCf  = [ -4.220000e-26 -5.870000e-08  8.920000e-22
        5.950000e-08 -1.240000e-26 -1.275284e-03
        5.940000e-28  3.824210e-04 -4.240000e-24 ]
vCm  = [ -4.290000e-31  5.950000e-08 -3.390000e-26
       -5.870000e-08  3.540000e-24  3.824210e-04
        6.670000e-29 -1.275284e-03  1.940000e-26 ]
wCm  = [  1.969914e-03 -1.120000e-21  9.440000e-09
       -9.370000e-29  1.791046e-03 -2.440000e-26
        9.440000e-09  5.300000e-24  7.513490e-04 ]
```

- mass – lumped mass for the link in kilograms.
- cm_x – x coordinate of the center of mass for the link.
- cm_y – y coordinate of the center of mass for the link.
- cm_z – z coordinate of the center of mass for the link.
- vCf – 9 values representing the 3x3 vCf component of the compliance matrix. These are in row-major order.
- wCf – 9 values representing the 3x3 wCf component of the compliance matrix. These are in row-major order.

- vCm – 9 values representing the 3x3 vCm component of the compliance matrix. These are in row-major order.
- wCm – 9 values representing the 3x3 wCm component of the compliance matrix. These are in row-major order.

5.1.3.26 RA_DEFL_LINK_4_STOWED – Wrist (w/ Grapple stowed)

```
Parameter Set: RA_DEFL_LINK_4_STOWED
mass =  0.1481
cm_x =  0.0114
cm_y = -0.0673
cm_z = -0.008
vCf  =  [  5.710000e-09 -2.160000e-17  1.840000e-18
        -2.160000e-17  5.710000e-09  2.650000e-18
         1.840000e-18  2.650000e-18  5.710000e-09 ]
wCf  =  [ -1.220000e-35 -6.090000e-13 -2.140000e-10
         1.830000e-11 -2.150000e-37  1.490000e-10
         2.140000e-10 -4.960000e-12  9.290000e-36 ]
vCm  =  [ -1.160000e-35  1.830000e-11  2.140000e-10
        -6.090000e-13  2.340000e-36 -4.960000e-12
        -2.140000e-10  1.490000e-10  1.240000e-35 ]
wCm  =  [  4.930000e-05 -9.850000e-28 -0.000000e+00
        -9.850000e-28  1.475213e-03  1.230000e-28
         1.280000e-31  9.770000e-29  4.930000e-05 ]
```

- mass – lumped mass for the link in kilograms.
- cm_x – x coordinate of the center of mass for the link.
- cm_y – y coordinate of the center of mass for the link.
- cm_z – z coordinate of the center of mass for the link.
- vCf – 9 values representing the 3x3 vCf component of the compliance matrix. These are in row-major order.
- wCf – 9 values representing the 3x3 wCf component of the compliance matrix. These are in row-major order.
- vCm – 9 values representing the 3x3 vCm component of the compliance matrix. These are in row-major order.
- wCm – 9 values representing the 3x3 wCm component of the compliance matrix. These are in row-major order.

5.1.3.27 RA_DEFL_LINK_4_UNSTOWED – Wrist (w/ Grapple Unstowed)

```
Parameter Set: RA_DEFL_LINK_4_UNSTOWED
mass =  0.6391
cm_x =  0.0026
cm_y = -0.0156
cm_z =  0.0323
vCf  =  [  5.710000e-09 -2.160000e-17  1.840000e-18
        -2.160000e-17  5.710000e-09  2.650000e-18
         1.840000e-18  2.650000e-18  5.710000e-09 ]
wCf  =  [ -1.220000e-35 -6.090000e-13 -2.140000e-10
         1.830000e-11 -2.150000e-37  1.490000e-10
```

```
        2.140000e-10 -4.960000e-12  9.290000e-36 ]
vCm  = [ -1.160000e-35  1.830000e-11  2.140000e-10
       -6.090000e-13  2.340000e-36 -4.960000e-12
       -2.140000e-10  1.490000e-10  1.240000e-35 ]
wCm  = [  4.930000e-05 -9.850000e-28 -0.000000e+00
       -9.850000e-28  1.475213e-03  1.230000e-28
        1.280000e-31  9.770000e-29  4.930000e-05 ]
```

- mass – lumped mass for the link in kilograms.
- cm_x – x coordinate of the center of mass for the link.
- cm_y – y coordinate of the center of mass for the link.
- cm_z – z coordinate of the center of mass for the link.
- vCf – 9 values representing the 3x3 vCf component of the compliance matrix. These are in row-major order.
- wCf – 9 values representing the 3x3 wCf component of the compliance matrix. These are in row-major order.
- vCm – 9 values representing the 3x3 vCm component of the compliance matrix. These are in row-major order.
- wCm – 9 values representing the 3x3 wCm component of the compliance matrix. These are in row-major order.

5.1.3.28 RA_DEFL_LINK_5 – Tool

```
Parameter Set: RA_DEFL_LINK_5
mass =  0
cm_x =  0
cm_y =  0
cm_z =  0
vCf  =  [  0.000000e+00  0.000000e+00  0.000000e+00
        0.000000e+00  0.000000e+00  0.000000e+00
        0.000000e+00  0.000000e+00  0.000000e+00 ]
wCf  =  [  0.000000e+00  0.000000e+00  0.000000e+00
        0.000000e+00  0.000000e+00  0.000000e+00
        0.000000e+00  0.000000e+00  0.000000e+00 ]
vCm  =  [  0.000000e+00  0.000000e+00  0.000000e+00
        0.000000e+00  0.000000e+00  0.000000e+00
        0.000000e+00  0.000000e+00  0.000000e+00 ]
wCm  =  [  0.000000e+00  0.000000e+00  0.000000e+00
        0.000000e+00  0.000000e+00  0.000000e+00
        0.000000e+00  0.000000e+00  0.000000e+00 ]
```

- mass – lumped mass for the link in kilograms.
- cm_x – x coordinate of the center of mass for the link.
- cm_y – y coordinate of the center of mass for the link.
- cm_z – z coordinate of the center of mass for the link.
- vCf – 9 values representing the 3x3 vCf component of the compliance matrix. These are in row-major order.
- wCf – 9 values representing the 3x3 wCf component of the compliance matrix. These are in row-major order.

- vCm – 9 values representing the 3x3 vCm component of the compliance matrix. These are in row-major order.
- wCm – 9 values representing the 3x3 wCm component of the compliance matrix. These are in row-major order.

5.1.3.29 RA_THERMAL

```
Parameter Set: RA_THERMAL
temp_min      = [ -55     -55     -55     -55     -55 ]
temp_delta    = [  20      20      20      20      20 ]
temp_max_rotor = [  95      95      95      95         ]
temp_max_case  = [  75      75      75      75         ]
temp_cooldown  = [  45      45      45      45         ]
CM_r          = [  0.1014   0.1014   0.1014   0.1014    ]
R_rtc         = [  0.0885   0.0885   0.0885   0.0885    ]
I_nl          = [  0.008    0.008    0.008    0.008     ]
R_cal         = [  42      42      42      42         ]
T_cal         = [  20      20      20      20         ]
alpha         = [  0.00393  0.00393  0.00393  0.00393   ]
filter_alpha  =  0.05
filter_dt     =  900
filter_c_0    =  4.429
filter_c_1    =  0.154
filter_c_2    =  0
filter_c_3    =  0
filter_c_4    =  0
```

- temp_min – joint1, joint2, joint3, joint4, unused joint temperature (Celsius) below which the FSW will turn on heaters to warm up actuators during motion or during execution of the IDA_THERMOSTAT command. If an actuator is not up to temperature when a move command is issued, motion is not initiated until the joint warms up. If a temperature sensor or heater fails, set *temp_min* to -800 Celsius. Otherwise, motion will not be initiated.
- temp_delta - joint1, joint2, joint3, joint4, unused joint temperature delta (Celsius). Above *temp_min* + *temp_delta* the actuator heaters are turned off if they are on.
- temp_max_rotor - joint1, joint2, joint3, joint4 joint actuator rotor temperature upper limit (Celsius). Computed from case temperature using thermal model.
- temp_max_case - joint1, joint2, joint3, joint4 joint actuator case temperature upper limit (Celsius). Measured from PRTs.
- temp_cooldown - joint1, joint2, joint3, joint4 temperature to reach during cool down before resuming motion (Celsius).
- CM_r - joint1, joint2, joint3, joint4 rotor thermal mass (joules/ Celsius).
- R_rtc - joint1, joint2, joint3, joint4 1/(rotor-to-case thermal resistance) (watts/ Celsius).
- I_nl - joint1, joint2, joint3, joint4 motor no-load current (A).
- R_cal - joint1, joint2, joint3, joint4 rotor resistance at cal temp (ohms).
- T_cal - joint1, joint2, joint3, joint4 cal temp (Celsius).
- alpha - joint1, joint2, joint3, joint4 temperature coefficient of resistance of copper (ohms/Celsius).
- filter_alpha - weighting coefficient for ISAD rasp motor temperature filter.
- filter_dt - reseed the filter if it has been longer than *filter_dt* since the last measurement.
- filter_c_0 – ISAD rasp motor temperature correction 0th order coefficient.
- filter_c_1 - ISAD rasp motor temperature correction 1st order coefficient.

- filter_c_2 - ISAD rasp motor temperature correction 2nd order coefficient.
- filter_c_3 - ISAD rasp motor temperature correction 3rd order coefficient.
- filter_c_4 - ISAD rasp motor temperature correction 4th order coefficient.

5.1.3.30 RA_GRAPPLE_PARAMS

```
Parameter Set: RA_GRAPPLE_PARAMS
grapple_heater          =  1
max_temperature         =  106.0
overheat_persistence    =  2
limit_switch_persistence =  2
```

- grapple_heater - which grapple heater to use.
    - 1 - Grapple heater 1 only
    - 2 - Grapple heater 2 only
    - 3 - Both grapple heaters
- max temperature - maximum allowed grapple actuator temperature (Celsius).
- overheat_persistence - number of consecutive times a grapple temperature must exceed the *max_temperature* parameter before a grapple overheat fault is declared.
- limit_switch_persistence - number of consecutive times the fingers-open limit switch must read tripped before the grapple opening algorithm will consider the fingers fully open.

5.1.3.31 RA_ACTIVE_HOLD_PARAMS

```
Parameter Set: RA_ACTIVE_HOLD_PARAMS
active_hold       =  0
hold_time         =  5.0
move_time_multiple =  2.0
```

- active_hold - whether to do active hold moves. 1.0 = Yes. Any other value means not to do active hold.
- hold_time - number of seconds to actively hold the position after completing an active hold move.
- move_time_multiple - duration of each individual timed move in an active hold move, expressed as a multiple of the control loop time (loop_time parameter in the Control group).

5.1.3.32 PAYLOAD_COL_PARAMS

```
Parameter Set: PAYLOAD_COL_PARAMS
seis_q_(i,j,k,r) =  [  0.000000  0.000000  0.177944  0.984041 ]
seis_t_(x,y,z)   =  [ -0.408000 -0.319000 -0.334000           ]
wts_q_(i,j,k,r)  =  [  0.000000  0.000000  0.000000  1.000000 ]
wts_t_(x,y,z)    =  [ -0.996000 -0.459000 -0.261000           ]
hp3_q_(i,j,k,r)  =  [  0.000000  0.000000 -0.175796  0.984427 ]
hp3_t_(x,y,z)    =  [ -0.861000 -0.008000 -0.455000           ]
```

- seis_q_(i,j,k,r) – SEIS quaternion vector from the lander frame to the static SEIS collision link frame.
    - seis_q_i – x * sin(theta/2)
    - seis_q_j – y * sin(theta/2)

  - seis_q_k – z * sin(theta/2)
  - seis_q_r – cos(theta/2)
- seis_t_(x,y,z) – x, y, z translation component from the lander frame to the static SEIS collision link frame.
- wts_q_(i,j,k,r) – WTS quaternion vector from the lander frame to the static WTS collision link frame.
  - wts_q_i – x * sin(theta/2)
  - wts_q_j – y * sin(theta/2)
  - wts_q_k – z * sin(theta/2)
  - wts_q_r – cos(theta/2)
- wts_t_(x,y,z) – x, y, z translation component from the lander frame to the static WTS collision link frame.
- hp3_q_(i,j,k,r) – HP3 quaternion vector from the lander frame to the static HP3 collision link frame.
  - wts_q_i – x * sin(theta/2)
  - wts_q_j – y * sin(theta/2)
  - wts_q_k – z * sin(theta/2)
  - wts_q_r – cos(theta/2)
- hp3_t_(x,y,z) – x, y, z translation component from the lander frame to the static HP3 collision link frame.

5.1.4 IDA Position in Spacecraft Coordinates

The IDA position in spacecraft coordinates file contains the Cartesian coordinates (x_coord(mm), y_coord(mm), z_coord(mm)) of the Lander Mechanical frame origin relative to the Payload frame. The file name is: ida_sc_coordinates.csv

5.1.5 Engineering/Science High Priority (SciHi) Data

Science High Priority data is non-channelized telemetry science data that is sent to the *high* priority APID (87). The data is collected after the previous data dump command is executed. Note that the previous data dump command sent science data to telemetry on that day and cleared the buffer for new data. It contains detailed sensor data typically collected every sample period and is used for reconstruction of IDA performance. The timestamp represents the last time IDA MC data was received.

The Science High Priority (SciHi) data product is a .csv file with parameter values described in Table 12. In each description, a snapshot of the parameter from the .csv file in excel is shown. The RA_DOF is a set that represents the 4 degrees of freedom of the IDA: shoulder yaw (azimuth), shoulder (elevation), elbow, and wrist pitch. Assume in IDA frame.

Table 12 - Science data parameters

Name (in .csv)	Units	Description	Snapshot of .csv in Excel
time(s)	second	time readings recorded	#time(s) 42209.8
enc1(rad) enc2(rad) enc3(rad) enc4(rad)	radians	encoder counts [RA_DOF]	enc1(rad) enc2(rad) enc3(rad) enc4(rad) -2.84 -1.9231 2.3173 -0.0007

pot1(rad) pot2(rad) pot3(rad) pot4(rad)	radians	potentiometer measurement [RA_DOF]	pot1(rad) pot2(rad) pot3(rad) pot4(rad) -2.8391 -1.9224 2.3209 -0.0018
tcase1(C) tcase2(C) tcase3(C) tcase4(C)	Celsius	measured motor case temp [RA_DOF]	enc1(rad) enc2(rad) enc3(rad) enc4(rad) -2.84 -1.9231 2.3173 -0.0007
trotor1(C) trotor2(C) trotor3(C) trotor4(C)	Celsius	computed motor rotor temperature [RA_DOF]	trotor1(C) trotor2(C) trotor3(C) trotor4(C) -52.2227 -52.7207 -50.0872 -51.1683
tgrapple(C)	Celsius	motor temperature	tgrapple(C) -26.9399
cur1(mA) cur2(mA) cur3(mA) cur4(mA)	milliampere	motor currents [RA_DOF]	cur1(mA) cur2(mA) cur3(mA) cur4(mA) 0 -0.3125 -0.3125 -0.3125
volt1(V) volt2(V) volt3(V) volt4(V)	volt	motor voltages [RA_DOF]	volt1(V) volt2(V) volt3(V) volt4(V) 0 0 0 0
stat	n/a	bit-mapped status word (Table 13)	stat 0
tool	n/a	currently selected tool (Table 14)	tool 5
x_enc(m) y_enc(m) z_enc(m)	meter	location of tool as Cartesian coordinates in IDA frame using joint encoder readings	x_enc(m) y_enc(m) z_enc(m) 1.3633 0.0043 0.4935
th_enc(m)	radian	orientation of tool in IDA frame using joint encoder readings	th_enc(m) 1.5014
x_pot(m) y_pot(m) z_pot(m)	meter	location of tool as Cartesian coordinates in IDA frame using joint potentiometer readings	x_pot(m) y_pot(m) z_pot(m) 1.3585 0.0026 0.5034
th_pot(m)	radian	orientation of tool in IDA frame using joint potentiometer readings	th_pot(m) 1.5014
x_enc_defl(m) y_enc_defl(m) z_enc_defl(m)	meter	deflected location of tool as Cartesian coordinates in IDA frame using joint encoder readings	x_enc_defl(m) y_enc_defl(m) z_enc_defl(m) 1.3619 0.0032 0.497
th_enc_defl(m)	radian	deflected orientation of tool in IDA frame using joint encoder readings	th_enc_defl(m) 1.5014
x_pot_defl(m) y_pot_defl(m) z_pot_defl(m)	meter	deflected location of tool as Cartesian coordinates in IDA frame using joint potentiometer readings	x_pot_defl(m) y_pot_defl(m) z_pot_defl(m) 1.3571 0.0016 0.5068
th_pot_defl(m)	radian	deflected orientation of tool in IDA frame using	th_pot_defl(m) 1.5014

		joint potentiometer readings	
v1(rad/s) v2(rad/s) v3(rad/s) v4(rad/s)	radians per second	joint velocity [RA_DOF]	v1(rad/s) v2(rad/s) v3(rad/s) v4(rad/s) 0.0024 0.0006 0.001 0.0025
tau1(N-m) tau2(N-m) tau3(N-m) tau4(N-m)	Newton meter	joint torque [RA_DOF]	tau1(N-m) tau2(N-m) tau3(N-m) tau4(N-m) 0 14.5451 -41.0253 0
Fr(N) Ft(N) Fz(N)	Newton	end-effector force in tool frame	Fr(N) Ft(N) Fz(N) -3.6661 -0.7414 -0.0603
htr_cur(mA)	milliampere	sum of heater currents	htr_cur(mA) 0.0015
grapple_phase	n/a	grapple phase (Table 15)	grapple_phase 2

In Table 12, the computed motor rotor temperatures T_R (trotor1(C), trotor2(C), trotor3(C), trotor4(C)) are obtained

from the motor case temperature T_C (tcase1(C), tcase2(C), tcase3(C), tcase4(C)) using a thermal model. The IDA MC measures each motor case temperature T_C (tcase1(C), tcase2(C), tcase3(C), tcase4(C)) and reports them to the IDA FSW as part of the IDA sensor data response message. At each timestep, the IDA FSW computes each motor rotor temperature T_R (trotor1(C), trotor2(C), trotor3(C), trotor4(C)) using the thermal model described below.

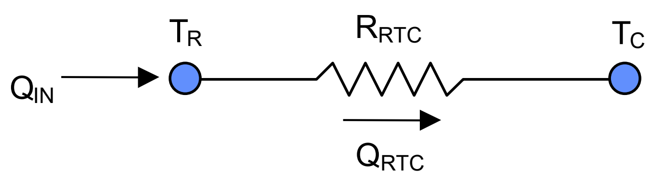


Figure 8 - Motor Temperature Model

$$Q_{IN} = I^2 R + I_{NL} V$$

$$Q_{RTC} = (T_R - T_C) R_{RTC}$$

$$R = R_{20C}(1 + T_{COEFF}(T_R - T_{CAL}))$$

$$T_R += (Q_{IN} - Q_{RTC}) * \Delta t * (CM_R)$$

Where

Q_{IN} is the heat load on the rotor,

I is the motor current in amps which is converted by the FSW software from the raw digital number value by the following formula: $I = I_M * I_H - I_B$

where I_M is the joint motor current slope,

I_H is the joint measured raw value,

I_B is the joint motor current offset (LSBs), current reference value when the joint angle is zero,

R is the rotor resistance in ohms,

I_{NL} is the no-load current in amps,

T_R is the joint rotor temperature in Celsius,

T_C is the joint case temperature in Celsius,

V is the motor voltage in volts,

R_{RTC} is 1/(rotor-to-case thermal resistance) in watts/Celsius,

R_{20C} is the rotor resistance at 20ºC in ohms,

T_{COEFF} is the temperature coefficient of resistance of copper in ohms/Celsius,

T_{CAL} is the calibration temperature in Celsius,

Δt is the actuator control loop rate in seconds (nominally 10Hz),

CM_R is the rotor thermal mass in Joules/Celsius,

+= is treated as the C programming assignment operator.

The parameter values of the thermal model are found in the IDA Parameters files RA_JOINT_PARAMS (section **Error! Reference source not found.**) and RA_THERMAL section 5.1.3.29).

The stat value is a word of 32 bits, which are described in Table 13. The bits are grouped in fours to represent the IDA 4 degrees of freedom. In each group of four bits, the very right bit corresponds to joint 1 (azimuth), the second to right bit to joint 2 (elevation), the third to right bit to joint 3 (elbow), and the very left bit to joint 4 (wrist).

The deflected location and orientation of tool as Cartesian coordinates in IDA frame using joint encoder readings (x_enc_defl(m), y_enc_defl(m), z_enc_defl(m), th_enc_defl(m)) and using joint potentiometer readings (x_pot_defl(m), y_pot_defl(m), z_pot_defl(m), th_pot_defl(m)) are obtained from, respectively, the location of tool using joint encoder readings (x_enc(m), y_enc(m), z_enc(m), th_enc(m)) and using joint potentiometer readings (x_pot(m), y_pot(m), z_pot(m), th_pot(m)). Due to the stiffness and mass of the IDA, the links can deflect under their own weight and the weight of a grappled payload. The IDA FSW computes the expected deflection from an internal model of the IDA stiffness and mass properties, knowledge of the grappled science instrument payload, payload masses, and the relative direction and magnitude of gravity [10][11]..

The end-effector forces (Fr(N), Ft(N), Fz(N)) are estimated from the joint motor current (cur1(mA), cur2(mA), cur3(mA), cur4(mA)) in the selected tool frame **Error! Reference source not ound.Error! Reference source not found.**. Fr(N) is the radial force, Ft(N) is the translational force that is perpendicular Fr(N), and Fz(N) is the vertical force. More details on the end effector forces model are provided in Appendix D.

Table 13 - Bit-mapped status word

Bits	**Description**	**0**	**1**
0-3	motor power	off	on
4-7	direction of rotation	pos	neg
8-11	motor over current	no	yes

12-15	brake	on	off
16-19	heaters	off	on
20-31	spares	N/A	N/A

Table 14 - Selected tool

Number	Tool
0	Blade 1
1	Blade 2
2	Scoop
3	IDC
4	Wrist
5	Grapple
6	Grapple-Attach
7	SEIS
8	WTS
9	HP3

Table 15 - Grapple phase

Number	Grapple Phase
0	Launch-Locked
1	Stowed
2	OK-To-Open
3	SEIS-Grappled
4	WTS-Grappled
5	HP3-Grappled

5.1.6 Engineering/Science Low Priority (SciLo) Data

Science Low Priority data is non-channelized telemetry science data that is sent to the *low* priority APID (88). The data and data product are equivalent to Science High Priority.

5.1.7 RSVP Replay MP4

An IDA engineer will create a MPEG-4 video of RSVP for the IDA activities performed during the soil mechanics experiments from the saved images from an RSVP simulation. The command is:

```
ffmpeg -i {inputs}.png -vcodec libx264 -pix_fmt yuv420p {output_name}.mp4
```

The resulting MP4 video will have H.264 video codec with no audio.

5.2 Document Product Formats

Documents in InSight archives are provided as PDF/A (https://www.pdfa.org/publication/iso-19005-pdfa/) or as plain ASCII text if no special formatting is required. Figures that accompany documents may be provided as TIFF, GIF, JPEG, or PNG files.

5.3 PDS Label

IDA raw and calibrated products are archived at the PDS Geosciences Node using the PDS4 archive standard [1-4]. The products are grouped into collections and bundles as described in Section 4. Each product is assigned a unique Logical Identifier (LID) and is accompanied by a label containing the product metadata. Labels are written in XML (Extensible Markup Language) and conform to XML and PDS4 standards. An example of an IDA label is given in Appendix E.

APPENDIX A – SUPPORT STAFF AND COGNIZANT PERSONS

Table 16 - Archive Support Staff and Cognizant Persons

IDA Team		
Name	**Affiliation**	**Email**
Hallie Abarca	JPL	hallie.e.gengl@jpl.nasa.gov
Eloise Marteau	JPL	eloise.marteau@jpl.nasa.gov
Ashitey Trebi-Ollennu	JPL	ashitey.trebi-ollennu@jpl.nasa.gov
Grace Lim	JPL	grace.lim@jpl.nasa.gov

PDS Geosciences Node		
Name	**Affiliation**	**Email**
Raymond Arvidson, director	Washington University	arvidson@wunder.wustl.edu
Edward Guinness	Washington University	guinness@wustl.edu
Susan Slavney	Washington University	susan.slavney@wustl.edu

APPENDIX B – IDA DENAVIT-HARTENBERG LINK FRAMES

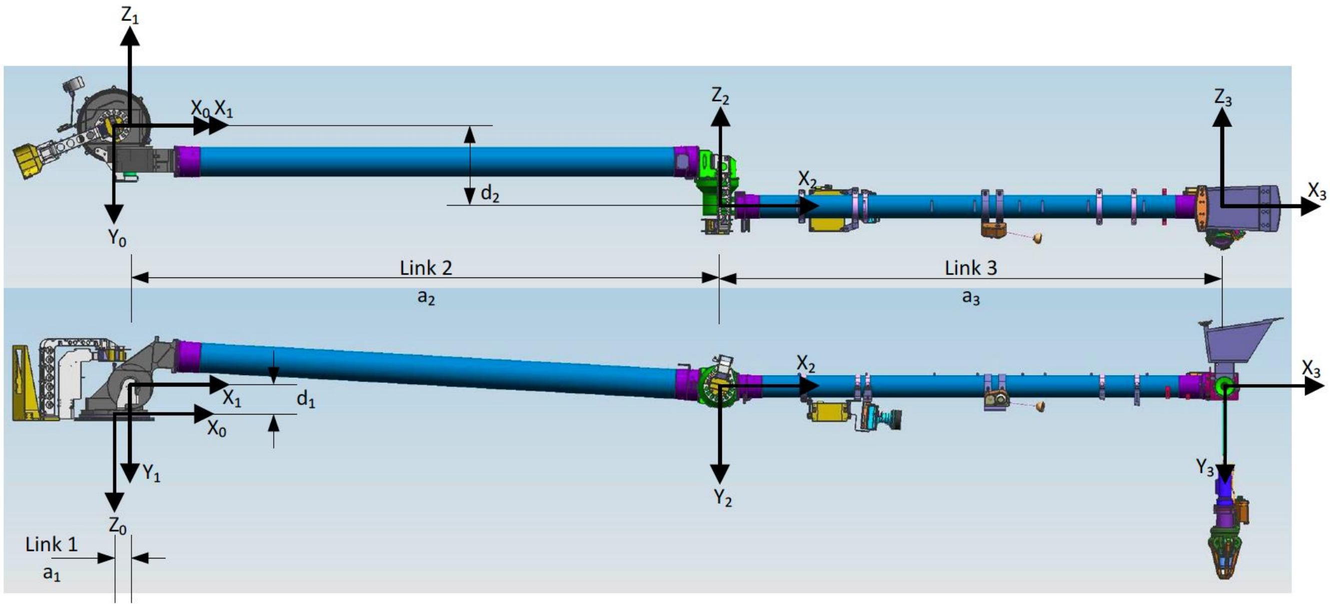


Figure 9 - Denavit-Hartenberg link frames

APPENDIX C – STRUCTURE OF THE IDA SCIENCE BLOCK

The IDA flight software produces a number of different data blocks which reflect the IDA status during various phases of its operations. The IDA data which is useful for science investigation is organized in a data block which is generated by the IDA FSW at a regular frequency. These blocks are stored on-board and are downlinked later during a telemetry pass.

The structure is shown in C language syntax:

```
struct arm_science_1
{
    unsigned short int  enc[4];         /* encoder counts                */
    unsigned short int  pot[4];         /* potentiometer voltages        */
    unsigned short int  tcase[4];       /* measured motor case temp      */
    unsigned short int  trotor[4];      /* computed motor rotor temp     */
    unsigned short int  mtr_current[4]; /* motor currents                */
    ra_set_volt_arg     mtr_volt;            /* motor voltages                */
    unsigned int        grapple_temp;        /* grapple temperature           */
    unsigned int        status;              /* bit-mapped status word        */
    unsigned short int  htr_current;         /* sum of heater currents        */
    int                 tool;                /* currently selected tool       */
    enum ra_grapple_phase  grapple_phase;    /* grapple phase     */
    double              time;                /* time readings were recorded  */
}
```

APPENDIX D – END EFFECTOR FORCES MODEL

The end-effector forces model computes the radial force F_R and vertical force F_Z by solving the following system of equations:

$$c = \begin{bmatrix} F_R \\ F_Z \end{bmatrix} = A^{-1}b$$

where the vector b is given by:

$$b_{[0]} = -m_3 p_3(t_x \sin(q_1 + q_2 + q_3) - t_z \cos(q_1 + q_2 + q_3)) - m_2 p_2(t_x \sin(q_1 + q_2 + q_3) - t_z \cos(q_1 + q_2 + q_3) + a_2 \sin(q_1 + q_2)) - m_1 p_1(t_x \sin(q_1 + q_2 + q_3) - t_z \cos(q_1 + q_2 + q_3) + a_2 \sin(q_1 + q_2) + a_1 \sin(q_1))$$

$$b_{[1]} = -m_3 p_3(t_x \cos(q_1 + q_2 + q_3) + t_z \sin(q_1 + q_2 + q_3)) + m_2 p_2(t_x \cos(q_1 + q_2 + q_3) + t_z \sin(q_1 + q_2 + q_3) + a_2 \cos(q_1 + q_2)) + m_1 p_1(t_x \cos(q_1 + q_2 + q_3) + t_z \sin(q_1 + q_2 + q_3) + a_2 \cos(q_1 + q_2) + a_1 \cos(q_1))$$

and the matrix A is given by:

$$A_{[0][0]} = p_3(t_x \sin(q_1 + q_2 + q_3) - t_z \cos(q_1 + q_2 + q_3))^2 + p_2(t_x \sin(q_1 + q_2 + q_3) - t_z \cos(q_1 + q_2 + q_3) + a_2 \sin(q_1 + q_2))^2 + p_1(t_x \sin(q_1 + q_2 + q_3) - t_z \cos(q_1 + q_2 + q_3) + a_2 \sin(q_1 + q_2) + a_1 \sin(q_1))^2$$

$$A_{[0][1]} = -p_3(t_x \cos(q_1 + q_2 + q_3) + t_z \sin(q_1 + q_2 + q_3)) (t_x \sin(q_1 + q_2 + q_3) - t_z \cos(q_1 + q_2 + q_3)) - p_2(t_x \cos(q_1 + q_2 + q_3) + t_z \sin(q_1 + q_2 + q_3) + a_2 \cos(q_1 + q_2)) (t_x \sin(q_1 + q_2 + q_3) - t_z \cos(q_1 + q_2 + q_3) + a_2 \sin(q_1 + q_2)) - p_1(t_x \cos(q_1 + q_2 + q_3) + t_z \sin(q_1 + q_2 + q_3) + a_2 \cos(q_1 + q_2) + a_1 \cos(q_1)) (t_x \sin(q_1 + q_2 + q_3) - t_z \cos(q_1 + q_2 + q_3) + a_2 \sin(q_1 + q_2) + a_1 \sin(q_1))$$

$$A_{[1][0]} = A_{[0][1]}$$

$$A_{[1][1]} = p_3(t_x \cos(q_1 + q_2 + q_3) + t_z \sin(q_1 + q_2 + q_3))^2 + p_2(t_x \cos(q_1 + q_2 + q_3) + t_z \sin(q_1 + q_2 + q_3) + a_2 \cos(q_1 + q_2))^2 + p_1(t_x \cos(q_1 + q_2 + q_3) + t_z \sin(q_1 + q_2 + q_3) + a_2 \cos(q_1 + q_2) + a_1 \cos(q_1))^2$$

where:

$q_{[0-3]}$ are the joint encoder values from SciHi Data enc1(rad), enc2(rad), enc3(rad), enc4(rad)) in Table 12,

$p_{[0-3]}$ is the joint motor power (1.0 if on, 0.0 if off),

$t_{\{x,y,z\}}$ is the wrist-to-tool vector found in **Error! Reference source not found.** A_ARM_PARAMS, tool from SciHi Data in Table 14

$a_{[0-3]}$ are the joint link lengths found in **Error! Reference source not found.** A_ARM_PARAMS,

$d_{[0-3]}$ are the joint link offsets found in **Error! Reference source not found.** A_ARM_PARAMS,

$m_{[0-3]}$ are the joint moments with gravity considered which is obtained as follows:

$$m_i = tau_i + moment_{iz}$$

$$tau_i = \; mtr\_torq\_sign_i * (mtr_{current_i} - \; cur_{no_{load}\,i}) * \; Ka_i$$

where:

$moment_{iz}$ is the z-component of moment at the RA frames of joint i,

$mtr\_torq\_sign = \{-1.0, 1.0, 1.0, 1.0\}$,

$mtr_{current_i}$ is the motor current of joint i,

$cur_{no_{load}\,i}$ is the no-load current for each joint as a function of temperature,

Ka_i is the torque constant for each actuator as a function of temperature

The transverse force F_T is obtained from:

$$F_T = \frac{\left(F_R\left(d_1 + t_y\right) + m_0 \right)}{\left(a_2\cos(q_1 + \; q_2) + a_1\cos(q_1) + a_0 + \; t_x\cos(q_1 + \; q_2 + \; q_3) + t_z\sin(q_1 + \; q_2 + \; q_3) \right)}$$

APPENDIX E – EXAMPLE PDS LABEL FOR AN IDA DATA PRODUCT


```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<?xml-model href="http://pds.nasa.gov/pds4/pds/v1/PDS4_PDS_1B10.sch"
  schematypens="http://purl.oclc.org/dsdl/schematron"?>
<?xml-model href="http://pds.nasa.gov/pds4/msn/v1/PDS4_MSN_1B00_1100.sch"
  schematypens="http://purl.oclc.org/dsdl/schematron"?>

<Product_Observational xmlns="http://pds.nasa.gov/pds4/pds/v1"
    xmlns:msn="http://pds.nasa.gov/pds4/msn/v1"
    xmlns:pds="http://pds.nasa.gov/pds4/pds/v1"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://pds.nasa.gov/pds4/pds/v1
https://pds.nasa.gov/pds4/pds/v1/PDS4_PDS_1B10.xsd http://pds.nasa.gov/pds4/msn/v1
http://pds.nasa.gov/pds4/msn/v1/PDS4_MSN_1B00_1100.xsd">
  <Identification_Area>

<logical_identifier>urn:nasa:pds:insight_ida:data_raw_calibrated:a0087_0209_615088547_
615088547_190629192905_eu</logical_identifier>
    <version_id>1.0</version_id>
    <title>IDA Science High Data Product</title>
    <information_model_version>1.11.1.0</information_model_version>
    <product_class>Product_Observational</product_class>
    <Citation_Information>
      <publication_year>2019</publication_year>
      <description>IDA Science High Data Product</description>
    </Citation_Information>
  </Identification_Area>
  <Observation_Area>
    <Time_Coordinates>
      <start_date_time>2019-06-29T13:56:40.702Z</start_date_time>
      <stop_date_time>2019-06-29T13:56:40.702Z</stop_date_time>
    </Time_Coordinates>
    <Investigation_Area>
      <name>InSight</name>
      <type>Mission</type>
      <Internal_Reference>

<lid_reference>urn:nasa:pds:context:investigation:mission.insight</lid_reference>
        <reference_type>data_to_investigation</reference_type>
        <comment>This is the PDS4 logical identifier for the InSight
mission.</comment>
      </Internal_Reference>
    </Investigation_Area>
    <Observing_System>
      <Observing_System_Component>
        <name>InSight Lander</name>
        <type>Spacecraft</type>
        <Internal_Reference>

<lid_reference>urn:nasa:pds:context:instrument_host:spacecraft.insight</lid_reference>
          <reference_type>is_instrument_host</reference_type>
          <comment>This is the PDS4 logical identifier for the InSight Lander
spacecraft.</comment>
        </Internal_Reference>
      </Observing_System_Component>
      <Observing_System_Component>
        <name>Instrument Deployment Arm</name>
        <type>Instrument</type>
```

```
        <Internal_Reference>
          <lid_reference>urn:nasa:pds:context:instrument:ida.insight</lid_reference>
          <reference_type>is_instrument</reference_type>
        </Internal_Reference>
      </Observing_System_Component>
    </Observing_System>
    <Target_Identification>
      <name>Mars</name>
      <type>Planet</type>
      <Internal_Reference>
        <lid_reference>urn:nasa:pds:context:target:planet.mars</lid_reference>
        <reference_type>data_to_target</reference_type>
      </Internal_Reference>
    </Target_Identification>
    <Discipline_Area>
      <msn:Mission_Information>
        <msn:mission_phase_name>SURFACE MISSION</msn:mission_phase_name>
        <msn:product_type_name>IDA_SciHi</msn:product_type_name>
        <msn:spacecraft_clock_start>615088547</msn:spacecraft_clock_start>
        <msn:spacecraft_clock_stop>615088547</msn:spacecraft_clock_stop>
        <msn:Surface_Mission>
          <msn:start_sol_number>209</msn:start_sol_number>
        </msn:Surface_Mission>
      </msn:Mission_Information>
    </Discipline_Area>
  </Observation_Area>
  <Reference_List>
    <Internal_Reference>

<lid_reference>urn:nasa:pds:insight_ida:document:insight_ida_bundle_sis</lid_reference
>
      <reference_type>data_to_document</reference_type>
    </Internal_Reference>
  </Reference_List>
  <File_Area_Observational>
    <File>
      <file_name>A0087_0209_615088547_615088547_190629192905_eu.csv</file_name>
      <local_identifier>file</local_identifier>
    </File>
    <Header>
      <offset unit="byte">0</offset>
      <object_length unit="byte">678</object_length>
      <parsing_standard_id>7-Bit ASCII Text</parsing_standard_id>
      <description>Column headings</description>
    </Header>
    <Table_Delimited>
      <local_identifier>TABLE</local_identifier>
      <offset unit="byte">678</offset>
      <parsing_standard_id>PDS DSV 1</parsing_standard_id>
      <records>927</records>
      <record_delimiter>Carriage-Return Line-Feed</record_delimiter>
      <field_delimiter>Comma</field_delimiter>
      <Record_Delimited>
        <fields>58</fields>
        <groups>0</groups>
        <Field_Delimited>
          <name>time</name>
          <data_type>ASCII_Real</data_type>
          <unit>s</unit>
          <description>time</description>
        </Field_Delimited>
        <Field_Delimited>
          <name>elapsed_time</name>
```

```
      <data_type>ASCII_Real</data_type>
      <unit>hr</unit>
      <description>time readings recorded</description>
    </Field_Delimited>
    <Field_Delimited>
      <name>enc1</name>
      <data_type>ASCII_Real</data_type>
      <unit>rad</unit>
      <description>Joint 1 encoder counts [RA_DOF]</description>
    </Field_Delimited>
    <Field_Delimited>
      <name>enc2</name>
      <data_type>ASCII_Real</data_type>
      <unit>rad</unit>
      <description>Joint 2 encoder counts [RA_DOF]</description>
    </Field_Delimited>
    <Field_Delimited>
      <name>enc3</name>
      <data_type>ASCII_Real</data_type>
      <unit>rad</unit>
      <description>Joint 3 encoder counts [RA_DOF]</description>
    </Field_Delimited>
    <Field_Delimited>
      <name>enc4</name>
      <data_type>ASCII_Real</data_type>
      <unit>rad</unit>
      <description>Joint 4 encoder counts [RA_DOF]</description>
    </Field_Delimited>
    <Field_Delimited>
      <name>pot1</name>
      <data_type>ASCII_Real</data_type>
      <unit>rad</unit>
      <description>Joint 1 potentiometer measurement [RA_DOF]</description>
    </Field_Delimited>
    <Field_Delimited>
      <name>pot2</name>
      <data_type>ASCII_Real</data_type>
      <unit>rad</unit>
      <description>Joint 2 potentiometer measurement [RA_DOF]</description>
    </Field_Delimited>
    <Field_Delimited>
      <name>pot3</name>
      <data_type>ASCII_Real</data_type>
      <unit>rad</unit>
      <description>Joint 3 potentiometer measurement [RA_DOF]</description>
    </Field_Delimited>
    <Field_Delimited>
      <name>pot4</name>
      <data_type>ASCII_Real</data_type>
      <unit>rad</unit>
      <description>Join 4 potentiometer measurement [RA_DOF]</description>
    </Field_Delimited>
    <Field_Delimited>
      <name>tcase1</name>
      <data_type>ASCII_Real</data_type>
      <unit>degC</unit>
      <description>Joint 1 measured motor case temp [RA_DOF]</description>
    </Field_Delimited>
    <Field_Delimited>
      <name>tcase2</name>
      <data_type>ASCII_Real</data_type>
      <unit>degC</unit>
      <description>Joint 2 measured motor case temp [RA_DOF]</description>
```

```
    &lt;/Field_Delimited&gt;
    &lt;Field_Delimited&gt;
      &lt;name&gt;tcase3&lt;/name&gt;
      &lt;data_type&gt;ASCII_Real&lt;/data_type&gt;
      &lt;unit&gt;degC&lt;/unit&gt;
      &lt;description&gt;Joint 3 measured motor case temp [RA_DOF]&lt;/description&gt;
    &lt;/Field_Delimited&gt;
    &lt;Field_Delimited&gt;
      &lt;name&gt;tcase4&lt;/name&gt;
      &lt;data_type&gt;ASCII_Real&lt;/data_type&gt;
      &lt;unit&gt;degC&lt;/unit&gt;
      &lt;description&gt;Joint 4 &lt;/description&gt;
    &lt;/Field_Delimited&gt;
    &lt;Field_Delimited&gt;
      &lt;name&gt;trotor1&lt;/name&gt;
      &lt;data_type&gt;ASCII_Real&lt;/data_type&gt;
      &lt;unit&gt;degC&lt;/unit&gt;
      &lt;description&gt;Joint 1 computed motor rotor temp [RA_DOF]&lt;/description&gt;
    &lt;/Field_Delimited&gt;
    &lt;Field_Delimited&gt;
      &lt;name&gt;trotor2&lt;/name&gt;
      &lt;data_type&gt;ASCII_Real&lt;/data_type&gt;
      &lt;unit&gt;degC&lt;/unit&gt;
      &lt;description&gt;Joint 2 computed motor rotor temp [RA_DOF]&lt;/description&gt;
    &lt;/Field_Delimited&gt;
    &lt;Field_Delimited&gt;
      &lt;name&gt;trotor3&lt;/name&gt;
      &lt;data_type&gt;ASCII_Real&lt;/data_type&gt;
      &lt;unit&gt;degC&lt;/unit&gt;
      &lt;description&gt;Joint 3 computed motor rotor temp [RA_DOF]&lt;/description&gt;
    &lt;/Field_Delimited&gt;
    &lt;Field_Delimited&gt;
      &lt;name&gt;trotor4&lt;/name&gt;
      &lt;data_type&gt;ASCII_Real&lt;/data_type&gt;
      &lt;unit&gt;degC&lt;/unit&gt;
      &lt;description&gt;Joint 4 computed motor rotor temp [RA_DOF]&lt;/description&gt;
    &lt;/Field_Delimited&gt;
    &lt;Field_Delimited&gt;
      &lt;name&gt;tgrapple&lt;/name&gt;
      &lt;data_type&gt;ASCII_Real&lt;/data_type&gt;
      &lt;unit&gt;degC&lt;/unit&gt;
      &lt;description&gt;grapple motor temperature&lt;/description&gt;
    &lt;/Field_Delimited&gt;
    &lt;Field_Delimited&gt;
      &lt;name&gt;cur1&lt;/name&gt;
      &lt;data_type&gt;ASCII_Real&lt;/data_type&gt;
      &lt;unit&gt;mA&lt;/unit&gt;
      &lt;description&gt;Joint 1 motor currents [RA_DOF]&lt;/description&gt;
    &lt;/Field_Delimited&gt;
    &lt;Field_Delimited&gt;
      &lt;name&gt;cur2&lt;/name&gt;
      &lt;data_type&gt;ASCII_Real&lt;/data_type&gt;
      &lt;unit&gt;mA&lt;/unit&gt;
      &lt;description&gt;Joint 2 motor currents [RA_DOF]&lt;/description&gt;
    &lt;/Field_Delimited&gt;
    &lt;Field_Delimited&gt;
      &lt;name&gt;cur3&lt;/name&gt;
      &lt;data_type&gt;ASCII_Real&lt;/data_type&gt;
      &lt;unit&gt;mA&lt;/unit&gt;
      &lt;description&gt;Joint 3 motor currents [RA_DOF]&lt;/description&gt;
    &lt;/Field_Delimited&gt;
    &lt;Field_Delimited&gt;
      &lt;name&gt;cur4&lt;/name&gt;
```

```
          <data_type>ASCII_Real</data_type>
          <unit>mA</unit>
          <description>Joint 4 motor currents [RA_DOF]</description>
        </Field_Delimited>
        <Field_Delimited>
          <name>volt1</name>
          <data_type>ASCII_Real</data_type>
          <unit>V</unit>
          <description>Joint 1 motor voltages [RA_DOF]</description>
        </Field_Delimited>
        <Field_Delimited>
          <name>volt2</name>
          <data_type>ASCII_Real</data_type>
          <unit>V</unit>
          <description>Joint 2 motor voltages [RA_DOF]</description>
        </Field_Delimited>
        <Field_Delimited>
          <name>volt3</name>
          <data_type>ASCII_Real</data_type>
          <unit>V</unit>
          <description>Joint 3 motor voltages [RA_DOF]</description>
        </Field_Delimited>
        <Field_Delimited>
          <name>volt4</name>
          <data_type>ASCII_Real</data_type>
          <unit>V</unit>
          <description>Joint 4 motor voltages [RA_DOF]</description>
        </Field_Delimited>
        <Field_Delimited>
          <name>stat</name>
          <data_type>UTF8_String</data_type>
          <unit>none</unit>
          <description>bit-mapped status word</description>
        </Field_Delimited>
        <Field_Delimited>
          <name>tool</name>
          <data_type>ASCII_Integer</data_type>
          <unit>none</unit>
          <description>currently selected tool</description>
        </Field_Delimited>
        <Field_Delimited>
          <name>x_enc</name>
          <data_type>ASCII_Real</data_type>
          <unit>m</unit>
          <description>X location of tool as Cartesian coordinates in IDA frame using
joint encoder readings</description>
        </Field_Delimited>
        <Field_Delimited>
          <name>y_enc</name>
          <data_type>ASCII_Real</data_type>
          <unit>m</unit>
          <description>Y location of tool as Cartesian coordinates in IDA frame using
joint encoder readings</description>
        </Field_Delimited>
        <Field_Delimited>
          <name>z_enc</name>
          <data_type>ASCII_Real</data_type>
          <unit>m</unit>
          <description>Z location of tool as Cartesian coordinates in IDA frame using
joint encoder readings</description>
        </Field_Delimited>
        <Field_Delimited>
          <name>th_enc</name>
```

```
          <data_type>ASCII_Real</data_type>
          <unit>m</unit>
          <description>orientation of tool in IDA frame using joint encoder
readings</description>
        </Field_Delimited>
        <Field_Delimited>
          <name>x_pot</name>
          <data_type>ASCII_Real</data_type>
          <unit>m</unit>
          <description>x location of tool as Cartesian coordinates in IDA frame using
joint potentiometer readings</description>
        </Field_Delimited>
        <Field_Delimited>
          <name>y_pot</name>
          <data_type>ASCII_Real</data_type>
          <unit>m</unit>
          <description>y location of tool as Cartesian coordinates in IDA frame using
joint potentiometer readings</description>
        </Field_Delimited>
        <Field_Delimited>
          <name>z_pot</name>
          <data_type>ASCII_Real</data_type>
          <unit>m</unit>
          <description>z location of tool as Cartesian coordinates in IDA frame using
joint potentiometer readings</description>
        </Field_Delimited>
        <Field_Delimited>
          <name>th_pot</name>
          <data_type>ASCII_Real</data_type>
          <unit>m</unit>
          <description>orientation of tool in IDA frame using joint potentiometer
readings</description>
        </Field_Delimited>
        <Field_Delimited>
          <name>x_enc_defl</name>
          <data_type>ASCII_Real</data_type>
          <unit>m</unit>
          <description>x deflected location of tool as Cartesian coordinates in IDA
frame using joint encoder readings</description>
        </Field_Delimited>
        <Field_Delimited>
          <name>y_enc_defl</name>
          <data_type>ASCII_Real</data_type>
          <unit>m</unit>
          <description>y deflected location of tool as Cartesian coordinates in IDA
frame using joint encoder readings</description>
        </Field_Delimited>
        <Field_Delimited>
          <name>z_enc_defl</name>
          <data_type>ASCII_Real</data_type>
          <unit>m</unit>
          <description>z deflected location of tool as Cartesian coordinates in IDA
frame using joint encoder readings</description>
        </Field_Delimited>
        <Field_Delimited>
          <name>th_enc_defl</name>
          <data_type>ASCII_Real</data_type>
          <unit>m</unit>
          <description>deflected orientation of tool in IDA frame using joint encoder
readings</description>
        </Field_Delimited>
        <Field_Delimited>
          <name>x_pot_defl</name>
```

```
          <data_type>ASCII_Real</data_type>
          <unit>m</unit>
          <description>x deflected location of tool as Cartesian coordinates in IDA
frame using joint potentiometer readings</description>
        </Field_Delimited>
        <Field_Delimited>
          <name>y_pot_defl</name>
          <data_type>ASCII_Real</data_type>
          <unit>m</unit>
          <description>y deflected location of tool as Cartesian coordinates in IDA
frame using joint potentiometer readings</description>
        </Field_Delimited>
        <Field_Delimited>
          <name>z_pot_defl</name>
          <data_type>ASCII_Real</data_type>
          <unit>m</unit>
          <description>z deflected location of tool as Cartesian coordinates in IDA
frame using joint potentiometer readings</description>
        </Field_Delimited>
        <Field_Delimited>
          <name>th_pot_defl</name>
          <data_type>ASCII_Real</data_type>
          <unit>m</unit>
          <description>deflected orientation of tool in IDA frame using joint
potentiometer readings</description>
        </Field_Delimited>
        <Field_Delimited>
          <name>v1</name>
          <data_type>ASCII_Real</data_type>
          <unit>rad/s</unit>
          <description>joint 1 velocity [RA_DOF]</description>
        </Field_Delimited>
        <Field_Delimited>
          <name>v2</name>
          <data_type>ASCII_Real</data_type>
          <unit>rad/s</unit>
          <description>joint 2 velocity [RA_DOF]</description>
        </Field_Delimited>
        <Field_Delimited>
          <name>v3</name>
          <data_type>ASCII_Real</data_type>
          <unit>rad/s</unit>
          <description>joint 3 velocity [RA_DOF]</description>
        </Field_Delimited>
        <Field_Delimited>
          <name>v4</name>
          <data_type>ASCII_Real</data_type>
          <unit>rad/s</unit>
          <description>joint 4 velocity [RA_DOF]</description>
        </Field_Delimited>
        <Field_Delimited>
          <name>tau1</name>
          <data_type>ASCII_Real</data_type>
          <unit>N-m</unit>
          <description>joint 1 torque [RA_DOF]</description>
        </Field_Delimited>
        <Field_Delimited>
          <name>tau2</name>
          <data_type>ASCII_Real</data_type>
          <unit>N-m</unit>
          <description>joint 2 torque [RA_DOF]</description>
        </Field_Delimited>
        <Field_Delimited>
```

```
          <name>tau3</name>
          <data_type>ASCII_Real</data_type>
          <unit>N-m</unit>
          <description>joint 3 torque [RA_DOF]</description>
        </Field_Delimited>
        <Field_Delimited>
          <name>tau4</name>
          <data_type>ASCII_Real</data_type>
          <unit>N-m</unit>
          <description>joint 4 torque [RA_DOF]</description>
        </Field_Delimited>
        <Field_Delimited>
          <name>Fr</name>
          <data_type>ASCII_Real</data_type>
          <unit>N</unit>
          <description>force at tool in cylindrical coordinates</description>
          <Special_Constants>
            <invalid_constant>-99999.0000</invalid_constant>
          </Special_Constants>
        </Field_Delimited>
        <Field_Delimited>
          <name>Ft</name>
          <data_type>ASCII_Real</data_type>
          <unit>N</unit>
          <description>force at tool in cylindrical coordinates</description>
          <Special_Constants>
            <invalid_constant>-99999.0000</invalid_constant>
          </Special_Constants>
        </Field_Delimited>
        <Field_Delimited>
          <name>Fz</name>
          <data_type>ASCII_Real</data_type>
          <unit>N</unit>
          <description>force at tool in cylindrical coordinates</description>
          <Special_Constants>
            <invalid_constant>-99999.0000</invalid_constant>
          </Special_Constants>
        </Field_Delimited>
        <Field_Delimited>
          <name>htr_cur</name>
          <data_type>ASCII_Real</data_type>
          <unit>mA</unit>
          <description>sum of heater currents</description>
        </Field_Delimited>
        <Field_Delimited>
          <name>grapple_phase</name>
          <data_type>ASCII_Integer</data_type>
          <unit>none</unit>
          <description>grapple phase</description>
        </Field_Delimited>
      </Record_Delimited>
    </Table_Delimited>
  </File_Area_Observational>
</Product_Observational>
```